Bridging images and natural language with deep learning

Gu, Jiuxiang

2019

Gu, J. (2019). Bridging images and natural language with deep learning. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/85399>

<https://doi.org/10.32657/10356/85399>

BRIDGING IMAGES AND NATURAL LANGUAGE WITH DEEP LEARNING

JIUXIANG GU

Interdisciplinary Graduate School

Rapid-Rich Object Search Lab

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2019

BRIDGING IMAGES AND NATURAL LANGUAGE WITH DEEP LEARNING

JIUXIANG GU

Interdisciplinary Graduate School

Rapid-Rich Object Search Lab

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the intellectual content of this thesis is the product of my original research work and has not been submitted for a higher degree to any other University or Institution.

2019-11-18

.....

Date

Gu Jiuxiang

.....

JIUXIANG GU

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

19 Nov. 2019

.....

Date

.....

Professor Jianfei Cai

Authorship Attribution Statement

This thesis contains material from 5 papers accepted at conferences and 1 paper published in the peer-reviewed journal where I am the first and corresponding author.

Chapter 3 is published as Jiuxiang Gu, Gang Wang, Jianfei Cai, Tsuhan Chen. An Empirical Study of Language CNN for Image Captioning. *IEEE International Conference on Computer Vision (ICCV)*, 2017.

The contributions of the co-authors are as follows:

- Dr. Gang Wang, Prof. Tsuhan Chen, Prof. Jianfei Cai, and I contributed to the idea discussions.
- I wrote the code and ran all experiments.
- I wrote the paper and rebuttal drafts. Dr. Gang Wang, Prof. Jianfei Cai, and Prof. Tsuhan Chen reviewed them and provided recommendations accordingly.

Chapter 4 is published as Jiuxiang Gu, Jianfei Cai, Gang Wang, Tsuhan Chen. Stack-Captioning: Coarse-to-Fine Learning for Image Captioning. *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

The contributions of the co-authors are as follows:

- Prof. Jianfei Cai, Dr. Gang Wang, Prof. Tsuhan Chen, and I contributed to idea discussions.
- I wrote the code and ran all experiments.
- I wrote the paper and rebuttal drafts. Prof. Jianfei Cai, Dr. Gang Wang, and Prof. Tsuhan Chen reviewed them and provided recommendations accordingly.

Chapter 5 is published as Jiuxiang Gu, Jianfei Cai, Joty Shafiq Rayhan, Li Niu, Gang Wang. Look, Imagine and Match: Improving Textual-Visual Cross-Modal Retrieval with Generative Models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

The contributions of the co-authors are as follows:

- Prof. Jianfei Cai, Prof. Joty Shafiq Rayhan, Dr. Li Niu, Dr. Gang Wang, and I contributed to idea discussions.
- I wrote the code and ran all experiments.
- I wrote the paper and rebuttal drafts. Prof. Jianfei Cai, Prof. Joty Shafiq Rayhan, Dr. Li Niu, and Dr. Gang Wang reviewed them and provided recommendations accordingly.

Chapter 6 is published as Jiuxiang Gu, Joty Shafiq Rayhan, Jianfei Cai, Gang Wang. Unpaired Image Captioning by Language Pivoting. *European Conference on Computer Vision (ECCV)*, 2018.

The contributions of the co-authors are as follows:

- Prof. Jianfei Cai, Prof. Joty Shafiq Rayhan, Dr. Gang Wang, and I contributed to idea discussions.
- I wrote the code and ran all experiments.
- I wrote the paper and rebuttal drafts. Prof. Joty Shafiq Rayhan, Prof. Jianfei Cai, and Dr. Gang Wang reviewed them and provided recommendations accordingly.

Chapter 7 is published as Jiuxiang Gu, Joty Shafiq Rayhan, Jianfei Cai, Handong Zhao, Xu Yang, Gang Wang. Unpaired Image Captioning via Scene Graph Alignments. *IEEE International Conference on Computer Vision (ICCV)*, 2019.

The contributions of the co-authors are as follows:

- Prof. Jianfei Cai, Prof. Joty Shafiq Rayhan, Dr. Handong Zhao, Mr. Xu Yang, Dr. Gang Wang and I contributed to idea discussions.
- I wrote the code and ran all experiments.
- Mr. Xu Yang assisted in checking the code.
- I wrote the paper draft. Prof. Joty Shafiq Rayhan, Prof. Jianfei Cai, Dr. Handong Zhao, Mr. Xu Yang, and Dr. Gang Wang reviewed them and provided recommendations accordingly.

2019-11-18

Date

JIUXIANG GU

Acknowledgements

First and foremost, I thank my supervisor, Prof. Jianfei Cai, for his precious guidance and grand support. I really appreciate his patient and generous advice, especially in discussing the details of research problems and improving my scientific writing. His academic attitude has influenced my entire Ph.D. study.

I am also grateful to Dr. Gang Wang, for encouraging me to pursue a Ph.D. degree. Not only is Dr. Wang is a great advisor in research, but he is also a mentor and friend. I would like to thank Prof. Tsuhan Chen for teaching me to think big and aim high. I would like to express my special thanks to Prof. Joty Shafiq Rayhan, for his patience and effort in helping me by providing me many insightful suggestions and making effective attempts on research. I am honored to have Prof. Sinno Jialin Pan and Prof. Xudong Jiang to serve on my doctoral committee. I am also very fortunate to work with Dr. Handong Zhao, Dr. Zhe Lin, Dr. Sheng Li, and Mr. Mingyang Ling during my internship at Adobe Research.

I would like to acknowledge the Interdisciplinary Graduate School (IGS), Nanyang Technological University (NTU), for providing me the NTU Research Scholarship to complete my Ph.D. study. This unique experience in NTU was so precious that I would cherish it forever. I would like to express my gratitude to my colleagues from Rapid-Rich Object Search (ROSE) Lab, for providing a comfortable research environment. I thank all of my friends and collaborators at NTU, for all their support of my life and research.

I would like to thank my parents, for providing unconditional love for me throughout my life. Their kindness and diligence personalities shaped my character. Many thanks to my sister, my brother-in-law, and my niece for their unconditional love and support. Finally, a heartfelt thanks goes out to my wife for all your love, support, and patience when I was only thinking about my research.

Abstract

We, as humans, can easily use our vision and language capabilities to accomplish a wide variety of tasks that combine the image and the text modalities. However, it is not easy for machines because it requires the model to understand the image and language, especially how they relate to each other. In recent years considerable progress has been made in applying deep learning to computer vision and natural language processing, but it is still challenging to connect images with natural language due to the different structures and characteristics between them.

In this thesis, I seek to bridge images and natural language with deep learning. Five different methods are proposed to reduce the gap between the image and the text modalities. They are convolutional neural network-based language model for image captioning, coarse-to-fine learning for image captioning, visual-textual cross-modal retrieval with generative models, unpaired image captioning by language pivoting, and unpaired image captioning via scene graph alignments. Overall, the major contributions of this thesis are as follows:

- A convolutional neural network-based language model which is suitable for statistical language modeling tasks is proposed in this thesis. This model is fed with all the previous words, while previous recurrent neural network-based language models predict the next word based on one previous word and the hidden state. Its ability of modeling the hierarchical structure and long-term information of words is critical for image captioning.
- A coarse-to-fine multi-stage prediction framework for image captioning is proposed in this thesis. The proposed coarse-to-fine approach is composed of multiple decoders, each of which operates on the output of the previous stage, producing increasingly refined image descriptions. Particularly, I optimize the model with a reinforcement learning approach which utilizes the output of each intermediate decoder's test-time inference algorithm as well as the output of its preceding decoder to normalize the rewards.

- A visual-textual cross-modal retrieval with generative learning is proposed in this thesis. Unlike existing cross-modal retrieval approaches that embed image-text pairs as single feature vectors in a common representational space, I propose to incorporate two generative processes (image-to-text and text-to-image) into the cross-modal feature embedding, through which the proposed model is able to learn not only the global abstract features but also the local grounded features.
- An unpaired image captioning via language pivoting is proposed in this thesis. I use a pivot language as an intermediary language to bridge the gap between an input image and a caption in the target language. The proposed method captures the characteristics of an image captioner from the pivot language and aligns it to the target language using another pivot-target sentence parallel corpus. In order to guide the target decoder to generate caption-like sentences, I have an autoencoder in the target language that guides the target language decoder to produce caption-like sentences.
- An unpaired image captioning via scene graph alignments is proposed in this thesis. The proposed framework comprises an image scene graph generator, a sentence scene graph generator, a scene graph encoder, and a sentence decoder. Specifically, I first train the scene graph encoder and the sentence decoder on the text modality. To align the scene graphs between images and sentences, I propose an unsupervised feature alignment method that maps the scene graph features from the image modality to the sentence modality without any paired training data.

Throughout the thesis, I demonstrate how each of the proposed methods can bridge the gap between images and natural language. Experimental results on public vision and language datasets have shown all these methods are able to obtain significant performance improvement on vision and language tasks such as image captioning and cross-modal retrieval.

Contents

Acknowledgements	ix
Abstract	xi
List of Figures	xvii
List of Tables	xxi
1 Introduction	1
1.1 Problem Statement	1
1.2 Challenges	4
1.3 Thesis Organization and Contributions	6
2 Literature Review	9
2.1 Deep Learning	9
2.1.1 Convolutional Neural Networks	9
2.1.2 Recurrent Neural Networks	16
2.2 Paired Image Captioning	18
2.2.1 Likelihood-Based Methods	19
2.2.2 Reinforcement Learning-Based Methods	21
2.2.3 Adversarial Learning-Based Methods	22
2.3 Unpaired Image Captioning	23
2.4 Cross-Modal Retrieval	24
3 An Empirical Study of Language CNN for Image Captioning	27
3.1 Introduction	27
3.2 Methodology	29
3.2.1 Framework	29
3.2.1.1 CNN _C Layer	30
3.2.1.2 Multimodal Fusion Layer	33
3.2.1.3 Recurrent Networks	33
3.2.2 Learning	35
3.3 Experiments	36
3.3.1 Datasets and Evaluation Metrics	36

3.3.2	Models	37
3.3.3	Implementation Details	38
3.3.3.1	Data Preprocessing	38
3.3.3.2	Training Details	38
3.3.3.3	Testing	39
3.3.4	Quantitative Results	39
3.3.4.1	Results Using CNN _L on MSCOCO	40
3.3.4.2	Results Using CNN _L on Flickr30K	42
3.3.4.3	Comparison with State-of-the-art Methods	42
3.3.5	Qualitative Results	44
3.4	Summary	45
4	Coarse-to-Fine Learning for Image Captioning	47
4.1	Introduction	47
4.2	Methodology	50
4.2.1	Image Encoding	50
4.2.2	Coarse-to-Fine Decoding	51
4.2.2.1	Coarse Decoder.	51
4.2.2.2	Fine Decoder.	52
4.2.2.3	Stacked Attention Model.	53
4.2.3	Learning	54
4.3	Experiments	57
4.3.1	Datasets and Setting	58
4.3.2	Baseline Approaches for Comparisons	58
4.3.3	Implementation Details	59
4.3.4	Quantitative Analysis	60
4.3.5	Qualitative Analysis	64
4.4	Summary	65
5	Improving Textual-Visual Cross-Modal Retrieval with Generative Models	67
5.1	Introduction	67
5.2	Methodology	70
5.2.1	System Overview	70
5.2.2	Cross-modal Feature Embedding	71
5.2.3	Image-to-text Generative Feature Learning	73
5.2.4	Text-to-image Generative Adversarial Feature Learning	74
5.3	Experiments	76
5.3.1	Dataset and Implementation Details	76
5.3.2	Baseline Approaches for Comparisons	78
5.3.3	Quantitative Results	79
5.3.3.1	Comparisons with the State-of-the-art	81
5.3.4	Qualitative Results	81
5.4	Summary	86

6	Unpaired Image Captioning by Language Pivoting	87
6.1	Introduction	87
6.2	Methodology	90
6.3	Experiments	96
6.3.1	Implementation Details	97
6.3.2	Quantitative Analysis	99
6.3.3	Qualitative Results	103
6.4	Summary	104
7	Unpaired Image Captioning via Scene Graph Alignment	105
7.1	Introduction	105
7.2	Method	108
7.2.1	Paired Image Captioning Revisited	108
7.2.2	Unpaired Image Captioning	109
7.2.2.1	Scene Graph Generator	110
7.2.2.2	Scene Graph Encoder	111
7.2.2.3	Sentence Decoder	113
7.2.2.4	Training and Inference	114
7.3	Experiments	117
7.3.1	Datasets and Setting	117
7.3.2	Implementation Details	118
7.3.3	Quantitative Results	120
7.3.4	Qualitative Results	125
7.4	Summary	126
8	Conclusions and Future Works	127
8.1	Conclusions	127
8.2	Future Works	128
8.2.1	Supervised Image Captioning	129
8.2.2	Cross-Modal Retrieval	129
8.2.3	Unpaired Image Captioning	130
	Author Publications	131
	Bibliography	133

List of Figures

1.1	A block diagram of simple encoder-decoder framework for image caption generation.	2
1.2	Conceptual differences between paired image captioning and unpaired image captioning.	3
1.3	Conceptual illustration of simple cross-modal feature embedding framework.	4
1.4	Conceptual illustration of visual-textual cross-modal retrieval. . . .	4
1.5	Illustration of the organization of the thesis.	6
2.1	The architecture of the LeNet-5 network, which works well on digit classification task.	12
2.2	The comparison of linear convolution layer and mlpconv layer. . . .	14
2.3	(A) Inception module, naive version. (B) The inception module used in [1]. (C) The improved inception module used in [2] where each 5×5 convolution is replaced by two 3×3 convolutions. (D) The Inception-ResNet-A module used in [3].	15
2.4	Illustration of simple RNN and its unfolded representation.	16
2.5	Conceptual illustration of paired image captioning.	19
3.1	An overview of our framework. The input of our model is a query image. Our model estimates the probability distribution of the next word given previous words and image. It consists of four parts: a CNN_I for image feature extraction, a deep CNN_L for language modeling, a multimodal layer (M) that connects the CNN_I and CNN_L , and a Recurrent Network (<i>e.g.</i> , RNN, LSTM, <i>etc.</i>) for word prediction. The weights are shared among all time frames.	29
3.2	The architecture of language CNN for sentence modeling. Here “/” stands for a zero padding. The CNN_L builds a hierarchical representation of history words that contains useful information for the next word prediction.	32
3.3	Illustration of (a) Simple RNN, and (b) RHN. $\tilde{r}$ is the hidden state, r is the new hidden state.	34
3.4	Qualitative results for images on MSCOCO. Ground-truth annotations (under each dashed line) and the generated annotations are shown for each image.	43

3.5	Some failure descriptions for images on MSCOCO, Ground-truth descriptions are under each dashed line.	43
4.1	Illustration of our proposed coarse-to-fine framework. Our model consists of one image encoder (CNN) and a sequence of sentence decoders (attention-based LSTM networks), and it takes the image as input and refines the image descriptions from coarse to fine. Here we show the increasingly improved image descriptions in two stages (gray and dark gray).	48
4.2	Illustration of the proposed coarse-to-fine decoding using intermediate supervision (reward) after each stage. The top row (gray) contains one coarse decoder (left) and two visual attention-based fine decoders under the training mode. The bottom row shows the fine decoders under its inference mode (greedy decoding) for computing rewards.	53
4.3	Visualizations of the generated captions and image attention maps on MSCOCO. Ground-Truth (GT) descriptions and the generated description of each stage are shown for each example. The columns from left to right correspond to the outputs of the three LSTM decoders from coarse to fine (coarse: <i>black</i> , refined: <i>purple</i> , final: <i>red</i>).	64
5.1	Conceptual illustration of our proposed cross-modal feature embedding with generative models. The cross-modal retrievals (<i>Image-to-Text</i> and <i>Text-to-Image</i>) are shown in different color. The two blue boxes are cross-modal data, and the generated data are shown in two dashed yellow clouds.	68
5.2	The proposed generative cross-modal learning framework (GXN). The entire framework consists of three training paths: cross-modal feature embedding (the entire upper part), image-to-text generative feature learning (the blue path), and text-to-image generative adversarial feature learning (the green path). It includes six networks: two sentence encoders RNN_{Enc}^h (dark green) and RNN_{Enc}^l (light green), one image encoder CNN_{Enc} (blue), one sentence decoder RNN_{Dec} , one image decoder CNN_{Dec} and one discriminator D_i	70
5.3	Visual results of image-to-text retrieval, where the top-5 retrieved captions and the generated caption are shown in red color.	82
5.4	Visual results of text-to-image retrieval. 2nd row: retrieved images. 3rd row: image samples generated by our conditional GAN.	82
5.5	More visual results of cross-modal retrieval.	83
5.6	Visualization of word embedding.	85

6.1	Pictorial depiction of our pivot-based unpaired image captioning setting. Here, i , x , y , and $\hat{y}$ denote source image, pivot language sentence, target language sentence, and ground truth captions in the target language, respectively. We use a dashed line to denote that there is no parallel corpus available for the pair. Solid lines with arrows represent decoding directions. Dashed lines inside a language (circle) denote stylistic and distributional differences between caption and translation data.	90
6.2	Illustration of our image captioning model with pivot language. The image captioning model first transforms an image into latent pivot sentences, from which our machine translation model generates the target caption.	94
6.3	Examples of the generated sentences on MSCOCO test images, where $i2t_{im \rightarrow zh}$ is the image captioner trained on AIC-ICC, $i2t_{im \rightarrow en}$ is the image captioner trained on MSCOCO, $i2t_{im \rightarrow zh \rightarrow en}(\mathcal{R}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}})$ and $i2t_{im \rightarrow zh \rightarrow en}(\mathcal{R}_{i \rightarrow x, x \rightarrow y})$ are our proposed models for unpaired image captioning, and GT stands for ground truth caption.	103
7.1	Illustration of our graph-based learning method. Our model consists of one visual scene graph detector (Top-Left), one fixed off-the-shelf scene graph language parser (Bottom-Left), a scene graph encoder G_{Enc}^S , a sentence decoder G_{Dec}^S , and a feature mapping module. . . .	106
7.2	The architectures for scene graph encoding, attention, and sentence decoding; g_{Att}^o , g_{Att}^r , and g_{Att}^a are attention modules for each kind of features, respectively.	113
7.3	Conceptual illustration of our unpaired feature mapping. For each kind of embedding p , there are two mapping functions $F_{I \rightarrow S}^p$ and $F_{S \rightarrow I}^p$, and two associated adversarial discriminators D_I^p and D_S^p . . .	115
7.4	Visualization of features in 2D space by t-SNE [4]. We plot the scatter diagrams for 1,500 samples.	124
7.5	Qualitative examples of different methods. In each example, the left image is the original input image; the middle is the image scene graph; the right image is the ground-truth sentence scene graph for compassion.	125
7.6	Examples of unpaired image captioning failure cases. Although the accuracy of image scene graph highly influences the performance of captioning results, our Graph-Align can still generate relevant image captions.	125

List of Tables

3.1	Performance comparison on MSCOCO, where M is short for METEOR, and S is short for SPICE (Bold numbers are the best results).	39
3.2	Performance comparison on Flickr30k (Bold numbers are the best results).	40
3.3	Performance in terms of BLEU- n , METEOR, and CIDEr compared with other state-of-the-art methods on the MSCOCO and Flickr30k datasets. For those competing methods, we extract their performance from their latest version of paper.	41
4.1	Performance comparisons on MSCOCO, where B@ n is short for BLEU- n , M is short for METEOR, and C is short for CIDEr. All values are reported as the percentage (Bold numbers are the best results).	60
4.2	Performance comparisons with the baselines on MSCOCO Karpathy test split. Our Stack-Cap (C2F) model achieves significant gains across all metrics.	61
4.3	Comparisons of the image captioning performance of the existing methods on MSCOCO Karpathy test split. Our Stack-Cap (C2F) model with the coarse-to-fine learning achieves significant gains across all metrics.	62
4.4	Leaderboard of the published image captioning models (as of 10/09/2017) on the online MSCOCO test server. Our single Stack-Cap model trained with the coarse-to-fine learning yields comparable performance with the state-of-the-art approaches on all reported metrics.	63
5.1	Cross-modal retrieval results on MSCOCO 1K-image test set (bold numbers are the best results).	80
5.2	Evaluating the quality of the retrieved captions on the MSCOCO 1K test set using the sentence-level metrics, where B@ n is a short form for BLEU- n , and C is a short form for CIDEr. All values are reported in percentage. The 2nd column is the rank order of the retrieved caption.	83
5.3	Comparisons of the cross-modal retrieval results on MSCOCO dataset with the state-of-the-art methods. We mark the unpublished work with * symbol. Note that ‘Sum’ is the summation of the two R@1 scores and the two R@10 scores.	84

5.4	Experimental results on Flickr30K 1k image test set.	85
6.1	Statistics of the datasets used in our experiments, where “im” denotes the image, “zh” denotes Chinese, and “en” denotes English. . .	96
6.2	Performance comparisons on AIC-ICC. The results of $i2t_{im \rightarrow zh}$ are achieved via beam search.	100
6.3	Performance comparisons on AIC-MT test dataset. Note that our $nmt_{zh \rightarrow en}$ model uses beam search.	100
6.4	Results of unpaired image-to-English captioning on MSCOCO 5K and Flickr30K 1K test splits, where M is short for METEOR. . . .	101
6.5	Self-BLEU scores on MSCOCO 5K test split. Note that lower Self-BLEU scores imply higher diversity of the image descriptions. . . .	102
6.6	Evaluation results of user assessment on MSCOCO 1.2K test split. .	103
7.1	Statistics of the training datasets.	118
7.2	Results for different sentence scene graph decoders on MSCOCO test split, where B@n refers to BLEU-n, M refers to METEOR, and C refers to CIDEr. All values are reported in percentage (bold numbers are the best results).	120
7.3	Results for different baselines without GAN training on the test split of the MSCOCO.	120
7.4	Ablation studies of different GAN losses for Graph-Align model. . .	121
7.5	The performances of using different feature mappings on MSCOCO test split. Shared GAN learns a shared feature mapping for three sets of features with CycleGAN. Single GAN concatenates the three kinds of embeddings together and learns a mapping with CycleGAN. .	122
7.6	Performance comparisons on the test split of the MSCOCO dataset. .	123

Chapter 1

Introduction

Leveraging on the rapid growth in the amount of the annotated data and the great improvements in the strengths of graphics processor units (GPUs), the research on deep learning has led to the tremendous advancement of computer vision [5, 6] and natural language processing (NLP) [7, 8]. Although both of the fields have flourished, their researches are not benefiting each other very well. More importantly, there are many real-world applications (*e.g.*, helping visually impaired people better understand their surroundings [9]) requiring knowledge from both of the fields. Hence, it is necessary to explore the possibility of integrating the two fields [10–14].

1.1 Problem Statement

The primary focus of this thesis is on bridging images and natural language with deep learning. In the following sub-sections, we will give a brief introduction of the two main problems investigated in this thesis: image captioning and cross-modal retrieval.

Image Captioning

Image captioning is a multimodal task involving computer vision and NLP. It aims to learn a mapping from image to natural language description. The mapping function can be learned with/without image-caption pairs. Ideally, a high-performing image caption generator should not only describe what is visually depicted in the image, but also generate appropriate or human-like sentences in the target language.

FIGURE 1.1: A block diagram of simple encoder-decoder framework for image caption generation.

Motivated by the recent advances in deep learning [5] and the release of large-scale datasets [15–17], many studies [18–24] have proposed to use the encoder-decoder framework for image caption generation. Figure 1.1 shows a simple block diagram of the encoder-decoder framework. The encoding-decoding process can be divided into two steps: first the image is encoded to features with an image encoder, then the image features are decoded to a sentence in the target language.

In this thesis, we investigate the image captioning problem from two aspects:

- **Paired setting.** As shown in Figure 1.2a, the goal of paired image captioning is to learn a model to describe an image with a caption with paired training data. The mapping function is trained in a supervised learning manner based on manually labeled image-caption pairs.
- **Unpaired setting.** As shown in Figure 1.2b, in the unpaired image captioning setting, we do not have ground-truth image-caption training pairs.

The goal of unpaired image captioning is to learn an image captioning model without paired training data.

FIGURE 1.2: Conceptual differences between paired image captioning and unpaired image captioning.

Cross-Modal Retrieval

As we are entering the era of big data, data from different modalities such as text, image, and video are growing at an unprecedented rate. Such multi-modal data exhibits heterogeneous properties, making it difficult for users to search information of interest effectively and efficiently. The target of cross-modal retrieval is to retrieve one modality at the output given the other modality at the input.

Figure 1.3 illustrates the common way for visual-textual cross-modal embedding, we first encode images and sentences into their respective features, and then map them into a common space, which is often optimized via a ranking loss that encourages the similarity of the mapped features of ground-truth image-text pairs to be greater than that of any other negative pair. Once the common representation is obtained, the similarity between the two modalities can be easily measured by computing distance between their representations in the common space.

FIGURE 1.3: Conceptual illustration of simple cross-modal feature embedding framework.

More specifically, as shown in Figure 1.4, the task of visual-textual cross-modal retrieval is to retrieve the images (resp. texts) that are relevant to a given textual (resp. image) query.

FIGURE 1.4: Conceptual illustration of visual-textual cross-modal retrieval.

1.2 Challenges

Image Captioning (Paired Setting)

Although deep learning models have dominated recent image caption generation tasks, effectively utilize the image information and generate accurate image descriptions is still challenging.

Firstly, most image captioning models follow the encoding-decoding framework [25] which consists of a Convolutional Neural Network (CNN) based image encoder

and a Recurrent Neural Network (RNN) based sentence decoder. However, vanilla RNNs-based approaches are limited in modeling the hierarchical structure and long-term dependencies in word sequences [26]. Although models like Long-Short Term Memory (LSTM) networks have memory cells which aim to memorize history information for a long term, they are still limited to several time steps because long-term information is gradually diluted at every time step [27].

Secondly, most of the existing image captioning works mainly focus on one-stage generation [18, 20, 28], but it is challenging to generate a rich image description in one stage. This is because rich descriptions require high-complexity models, where the problem of vanishing gradients often occurs, considering the back-propagated gradients diminish in strength as they propagate through many layers of a complex network.

Image Captioning (Unpaired Setting)

Compared with paired image captioning, the critical challenge in unpaired image captioning is that we do not have the ground-truth image-caption pairs during training. Unlike unpaired sentence-to-sentence [29] and image-to-image [30] translations where the encoders can be shared across the source and the target domains, unpaired image-to-sentence translation is more challenging because of the significantly different structures and characteristics between the image and text modalities. Besides, another challenge in unpaired image captioning is to fit the encoder-decoder framework.

Cross-Modal Retrieval

The fundamental challenge in cross-modal retrieval lies in the heterogeneity of different modalities of data. Thus, the learning of appropriate representations

for multi-modal data is crucial for cross-modal retrieval performance. Most of the existing studies on cross-modal textual-visual retrieval mainly focus on mapping the images and sentences to a common space with ranking loss. Although the feature representations in the learned common representation space have been successfully used to describe high-level semantic concepts of multi-modal data, they are not sufficient to retrieve images with detailed local similarity (*e.g.*, spatial layout) or sentences with word-level similarity.

1.3 Thesis Organization and Contributions

FIGURE 1.5: Illustration of the organization of the thesis.

Figure 1.5 shows the organization of this thesis. In Chapter 1, we give an introduction to this thesis and describe the problems and challenges. In Chapter 2, we present a literature review on image captioning methods and visual-textual cross-modal retrieval methods. In Chapter 3 and Chapter 4, we present the language-CNN based method and coarse-to-fine learning method for paired image captioning. In Chapter 5, we present the cross-modal retrieval which adopts image captioning as one generative learning path. In Chapter 6 and Chapter 7, we present the pivot

language-based method and the scene graph-based method for unpaired image captioning. Lastly, in Chapter 8, conclusions and future works are provided.

More specifically, we list the major contributions of the works in Chapter 3, Chapter 4, Chapter 5, Chapter 6, and Chapter 7 as follows:

Chapter 3. In this chapter, a CNN-based language model for image captioning is proposed. The proposed language CNN is fed with all the previous words and it can model the hierarchical structure and long-term dependencies in word sequences, which are critical for image captioning. This work has been published in the IEEE International Conference on Computer Vision, 2017 [31].

Chapter 4. In this chapter, a coarse-to-fine multi-stage prediction framework for image captioning is proposed. To address the exposure bias problem and the loss-evaluation mismatch problem, we optimize our model with a reinforcement learning approach which utilizes the output of each intermediate decoder's test-time inference algorithm as well as the output of its preceding decoder to normalize the rewards. This work has been published in the Thirty-Second AAAI Conference on Artificial Intelligence, 2018 [11].

Chapter 5. In this chapter, a generative cross-modal feature learning framework for visual-textual cross-modal retrieval is proposed. The proposed framework incorporates the image-to-text and the text-to-image generative models into the conventional cross-modal feature embedding, through which we are able to learn not only the global abstract features but also the local grounded features. This work has been published in the IEEE Conference on Computer Vision and Pattern Recognition, 2018 [13].

Chapter 6. In this chapter, a pivot language-based solution for unpaired image captioning is proposed. The proposed method couples an image-to-pivot captioning model with a pivot-to-target neural machine translation model in a joint learning framework. The coupling is done by adapting the word representations in

the encoder and the decoder of the neural machine translation model to produce caption-like sentences. This work has been published in the European Conference on Computer Vision, 2018 [12].

Chapter 7. In this chapter, a scene graph-based approach for unpaired image captioning is proposed. The proposed framework adopts the scene graph as an explicit representation to bridge the gap between image and sentence domains. In order to align the scene graphs between images and sentences, we propose an unsupervised feature alignment method which is able to map the scene graph features from the image to sentence modality without any paired training data. This work has been published in the IEEE International Conference on Computer Vision, 2019 [14].

Chapter 2

Literature Review

In this chapter, we present a literature review on deep learning, image captioning, and visual-textual cross-modal retrieval. First, we introduce the CNNs and RNNs. Then, we classify the paired image captioning methods into likelihood-based methods, reinforcement learning-based methods, and adversarial learning-based methods. After that, we focus on discussing the methods for unpaired image captioning. Finally, we review the methods for visual-textual cross-modal retrieval.

2.1 Deep Learning

2.1.1 Convolutional Neural Networks

Convolutional Neural Network is a well-known deep learning architecture inspired by the natural visual perception mechanism of the living creatures. In 1959, Hubel & Wiesel [32] found that cells in animal visual cortex are responsible for detecting light in receptive fields. Inspired by this discovery, Kunihiko Fukushima proposed the neocognitron in 1980 [33], which could be regarded as the predecessor of CNN. In 1990, LeCun *et al.* [?] published the seminal paper establishing the modern

framework of CNN, and later improved it in [34]. They developed a multi-layer artificial neural network called LeNet-5 which could classify handwritten digits. Like other neural networks, LeNet-5 has multiple layers and can be trained with the backpropagation algorithm [35]. It can obtain effective representations of the original image, which makes it possible to recognize visual patterns directly from raw pixels with little-to-none preprocessing. A parallel study of Zhang *et al.* [36] used a shift-invariant artificial neural network (SIANN) to recognize characters from an image. However, due to the lack of large training data and computing power at that time, their networks can not perform well on more complex problems, *e.g.*, large-scale image and video classification.

Since 2006, many methods have been developed to overcome the difficulties encountered in training deep CNNs [1, 37–39]. Most notably, Krizhevsky *et al.* [38] proposed a classic CNN architecture and showed significant improvements upon previous methods on the image classification task. The overall architecture of their method, *i.e.*, AlexNet [38], is similar to LeNet-5 but with a deeper structure. With the success of AlexNet, many works have been proposed to improve its performance. Among them, four representative works are ZFNet [40], VGGNet [39], GoogleNet [1] and ResNet [41]. From the evolution of the architectures, a typical trend is that the networks are getting deeper, *e.g.*, ResNet, which won the champion of ILSVRC 2015, is about 20 times deeper than AlexNet and 8 times deeper than VGGNet. By increasing depth, the network can better approximate the target function with increased nonlinearity and get better feature representations. However, it also increases the complexity of the network, which makes the network be more difficult to optimize and easier to get overfitting. Along this way, various methods have been proposed to deal with these problems in various aspects.

There are numerous variants of CNN architectures in the literature. However, their basic components are very similar. Taking the famous LeNet-5 as an example, it consists of three types of layers, namely convolutional, pooling, and fully-connected

layers. The convolutional layer aims to learn feature representations of the inputs. As shown in Figure 2.1, convolution layer is composed of several convolution kernels which are used to compute different feature maps. Specifically, each neuron of a feature map is connected to a region of neighbouring neurons in the previous layer. Such a neighbourhood is referred to as the neuron's receptive field in the previous layer. The new feature map can be obtained by first convolving the input with a learned kernel and then applying an element-wise nonlinear activation function on the convolved results. Note that, to generate each feature map, the kernel is shared by all spatial locations of the input. The complete feature maps are obtained by using several different kernels. Mathematically, the feature value at location (i, j) in the k -th feature map of l -th layer, $z_{i,j,k}^l$, is calculated by:

$$z_{i,j,k}^l = \mathbf{w}_k^{lT} \mathbf{x}_{i,j}^l + b_k^l \quad (2.1)$$

where $\mathbf{w}_k^l$ and b_k^l are the weight vector and bias term of the k -th filter of the l -th layer respectively, and $\mathbf{x}_{i,j}^l$ is the input patch centered at location (i, j) of the l -th layer. Note that the kernel $\mathbf{w}_k^l$ that generates the feature map $\mathbf{z}_{:, :, k}^l$ is shared. Such a weight sharing mechanism has several advantages such as it can reduce the model complexity and make the network easier to train. The activation function introduces nonlinearities to CNN, which are desirable for multi-layer networks to detect nonlinear features. Let $a(\cdot)$ denote the nonlinear activation function. The activation value $a_{i,j,k}^l$ of convolutional feature $z_{i,j,k}^l$ can be computed as:

$$a_{i,j,k}^l = a(z_{i,j,k}^l) \quad (2.2)$$

Typical activation functions are sigmoid, tanh [42] and ReLU [43]. The pooling layer aims to achieve shift-invariance by reducing the resolution of the feature maps. It is usually placed between two convolutional layers. Each feature map of a pooling layer is connected to its corresponding feature map of the preceding

FIGURE 2.1: The architecture of the LeNet-5 network, which works well on digit classification task.

convolutional layer. Denoting the pooling function as $\text{pool}(\cdot)$, for each feature map $\mathbf{a}_{:, :, k}^l$ we have:

$$y_{i,j,k}^l = \text{pool}(a_{m,n,k}^l), \forall (m,n) \in \mathcal{R}_{ij} \quad (2.3)$$

where $\mathcal{R}_{ij}$ is a local neighbourhood around location (i,j) . The typical pooling operations are average pooling [44] and max pooling [45].

After several convolutional and pooling layers, there may be one or more fully-connected layers which aim to perform high-level reasoning [39, 40, 46]. They take all neurons in the previous layer and connect them to every single neuron of current layer to generate global semantic information. Note that fully-connected layer not always necessary as it can be replaced by a 1×1 convolution layer [47].

The last layer of CNNs is an output layer. For classification tasks, the softmax operator is commonly used [38]. Another commonly used method is SVM, which can be combined with CNN features to solve different classification tasks [48, 49]. Let $\boldsymbol{\theta}$ denote all the parameters of a CNN (*e.g.*, the weight vectors and bias terms). The optimum parameters for a specific task can be obtained by minimizing an appropriate loss function defined on that task. Suppose we have N desired input-output relations $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}); n \in [1, \dots, N]\}$, where $\mathbf{x}^{(n)}$ is the n -th input data, $\mathbf{y}^{(n)}$ is its corresponding target label and $\mathbf{o}^{(n)}$ is the output of CNN. The loss of

CNN can be calculated as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N \ell(\boldsymbol{\theta}; \mathbf{y}^{(n)}, \mathbf{o}^{(n)}) \quad (2.4)$$

Training CNN is a problem of global optimization. By minimizing the loss function, we can find the best fitting set of parameters. Stochastic gradient descent is a common solution for optimizing CNN network [50, 51].

There have been various improvements on CNNs since the success of AlexNet in 2012. Network In Network (NIN) is a general network structure proposed by Lin *et al.* [47]. It replaces the linear filter of the convolutional layer by a micro network, *e.g.*, multilayer perceptron convolution (mlpconv) layer in the paper, which makes it capable of approximating more abstract representations of the latent concepts. The overall structure of NIN is the stacking of such micro networks. Figure 2.2 shows the difference between the linear convolutional layer and the mlpconv layer. Formally, the feature map of convolution layer (with nonlinear activation function, *e.g.*, ReLU [43]) is computed as:

$$a_{i,j,k} = \max(\mathbf{w}_k^T \mathbf{x}_{i,j} + b_k, 0) \quad (2.5)$$

where $a_{i,j,k}$ is the activation value of k -th feature map at location (i, j) , $\mathbf{x}_{i,j}$ is the input patch centered at location (i, j) , $\mathbf{w}_k$ and b_k are weight vector and bias term of the k -th filter. As a comparison, the computation performed by mlpconv layer is formulated as:

$$a_{i,j,k_n}^n = \max(\mathbf{w}_{k_n}^T \mathbf{a}_{i,j,:}^{n-1} + b_{k_n}, 0) \quad (2.6)$$

where $n \in [1, N]$, N is the number of layers in the mlpconv layer, $\mathbf{a}_{i,j,:}^0$ is equal to $\mathbf{x}_{i,j}$. In mlpconv layer, 1×1 convolutions are placed after the traditional convolutional layer. The 1×1 convolution is equivalent to the cross-channel parametric pooling operation which is succeeded by ReLU [43]. Therefore, the mlpconv layer

FIGURE 2.2: The comparison of linear convolution layer and mlpconv layer.

can also be regarded as the cascaded cross-channel parametric pooling on the normal convolutional layer. In the end, they also apply a global average pooling which spatially averages the feature maps of the final layer, and directly feed the output vector into softmax layer. Compared with the fully-connected layer, global average pooling has fewer parameters and thus reduces the overfitting risk and computational load.

Inception module is introduced by Szegedy *et al.* [1] which can be seen as a logical culmination of NIN. They use variable filter sizes to capture different visual patterns of different sizes, and approximate the optimal sparse structure by the inception module. Specifically, inception module consists of one pooling operation and three types of convolution operations (see Figure 2.3b), and 1×1 convolutions are placed before 3×3 and 5×5 convolutions as dimension reduction modules, which allow for increasing the depth and width of CNN without increasing the computational complexity. With the help of inception module, the network parameters can be dramatically reduced to 5 millions which are much less than those of AlexNet (60 millions) and ZFNet (75 millions).

FIGURE 2.3: (A) Inception module, naive version. (B) The inception module used in [1]. (C) The improved inception module used in [2] where each 5×5 convolution is replaced by two 3×3 convolutions. (D) The Inception-ResNet-A module used in [3].

In their later paper [2], to find high-performance networks with a relatively modest computation cost, they suggest the representation size should gently decrease from inputs to outputs as well as spatial aggregation can be done over lower dimensional embeddings without much loss in representational power. The optimal performance of the network can be reached by balancing the number of filters per layer and the depth of the network. Inspired by the ResNet [41], their latest Inception-V4 [3] combines the inception architecture with shortcut connections (see Figure 2.3d). They find that shortcut connections can significantly accelerate the training of inception networks. Their Inception-v4 model architecture (with 75 trainable layers) that ensembles three residual and one Inception-v4 can achieve 3.08% top-5 error rate on the validation dataset of ILSVRC 2012.

Acknowledgments. Section 2.1.1 is an edited reprint of the material as it appears in the following publication: Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Li Wang, Gang Wang, Jianfei Cai, Tsuhan Chen. Recent Advances in Convolutional Neural Networks. Pattern Recognition 77 (2018): 354-377. The dissertation author is the first and corresponding author.

2.1.2 Recurrent Neural Networks

Recurrent Neural Network (RNN) can be seen as an extension of the conventional feed-forward neural network, which is able to handle the variable-length sequence by having a recurrent hidden state h_t whose value at each time t is dependent on its previous state h_{t-1} . More formally, given a sequence $\{x_0, x_1, \dots, x_{T-1}\}$, the RNN updates its hidden state h_t as follows:

$$h_t = \phi(h_{t-1}, x_t) \quad (2.7)$$

where $\phi(\cdot)$ is usually an affine transformation followed by a non-linear activation function. Specifically, the update of the vanilla RNN hidden state in Eq. (2.7) can be implemented as:

$$h_t = g(\mathbf{W}x_t + \mathbf{U}h_{t-1}) \quad (2.8)$$

where $g(\cdot)$ is an activation function.

FIGURE 2.4: Illustration of simple RNN and its unfolded representation.

Figure 2.4 shows the simple RNN and its unfolded representation. Generally, the current state h_t will be fed into a softmax layer which outputs a probability distribution over the next element of the sequence. More formally, the sequence probability can be decomposed as:

$$p(x_0, \dots, x_{T-1}) = p(x_0)p(x_1 | x_0)p(x_2 | x_0, x_1) \cdots p(x_{T-1} | x_0, \dots, x_{T-2}) \quad (2.9)$$

where $p(x_t \mid x_0, \dots, x_{t-1})$ is the softmax output of h_t .

In practice, it is hard to train the vanilla RNN to capture long-term dependencies because the gradients tend to either vanish or explode [52]. To overcome the weakness, many RNN variants have been proposed.

Long Short-Term Memory unit [53] is a special type of RNN architecture designed to solve the vanishing gradient problem in RNNs [18, 23, 54]. A common LSTM unit is composed of a memory cell, an input gate, a forget gate, and an output gate. These gates allow the LSTM network capable of learning long-term dependencies. The transition equations of LSTM are formulated as:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ u_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(\mathbf{M}_{\text{lstm}} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \quad (2.10)$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \quad (2.11)$$

$$h_t = o_t \odot \tanh(c_t) \quad (2.12)$$

where $\odot$ indicates element-wise product, i_t is the *input* gate at time step t , f_t is the *forget* gate, o_t is the *output* gate, u_t denotes the modulated input, and c_t is the internal *memory cell* state. $\mathbf{M}_{\text{lstm}} : \mathbb{R}^{K+d} \rightarrow \mathbb{R}^{4d}$ is an affine transformation, where K is the dimensionality of input x_t and d is the dimension of hidden state h_{t-1} .

Gated Recurrent Unit (GRU) [55] is an architecture similar to the LSTM unit, but it has a simplified structure. GRU does not has a separate memory cell and exposes its hidden state without any control. The transition equations of GRU are

formulated as:

$$\begin{pmatrix} z_t \\ r_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \end{pmatrix} \left(\mathbf{M}_{\text{gru}} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \quad (2.13)$$

$$\tilde{h}_t = \tanh(\mathbf{W}_h(r_t \odot h_{t-1}) + \mathbf{U}_h x_t + \mathbf{b}_h) \quad (2.14)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (2.15)$$

where z_t is the *update* gate, and r_t is the *reset* gate. $\mathbf{M}_{\text{gru}} : \mathbb{R}^{K+d} \rightarrow \mathbb{R}^{2d}$ is an affine transformation. GRU is computationally more efficient and outperforms the LSTM network on many tasks [56] due to its simple structure.

Recurrent Highway Network (RHN) [57] introduces the highway connection to simple RNN. It has directly gated connections between previous state h_{t-1} and current input x_t to modulate the flow of information. The transition equations of RHN can be formulated as follows:

$$\begin{pmatrix} t_t \\ c_t \\ u_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(\mathbf{M}_{\text{rhn}} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \quad (2.16)$$

$$h_t = u_t \odot t_t + c_t \odot h_{t-1} \quad (2.17)$$

where t_t is the *transform* gate, c_t is the *carry* gate, and u_t denotes the modulated input. $\mathbf{M}_{\text{rhn}} : \mathbb{R}^{K+d} \rightarrow \mathbb{R}^{3d}$ is an affine transformation. Like GRU, RHN does not have *output* gate to control the exposure of the recurrent state, but exposes the whole state each time.

2.2 Paired Image Captioning

Most of the existing image captioning methods are under the paired setting, that is, each image has its corresponding ground-truth captions [11, 31]. As illustrated

in Figure 2.5, in paired setting, the training goal is to generate a caption from the input image such that the generated caption is similar to its ground-truth caption.

In this section, we classify the paired image captioning methods according to their training techniques: likelihood-based methods, reinforcement learning-based methods, and adversarial learning-based methods.

FIGURE 2.5: Conceptual illustration of paired image captioning.

2.2.1 Likelihood-Based Methods

Inspired by the success of the encoder-decoder framework for neural machine translation (NMT) [7, 58], many researchers have proposed to use such a framework for image caption generation [18, 20]. A lot of research works are typically trained by maximizing the likelihood of each ground-truth word given the previous ground-truth words and the image with back propagation [18–24]. During inference, the decoded word is drawn from the vocabulary according to the output probability at each time step.

One representative work in this direction is the method proposed by Vinyals *et al.* [18]. They encode the image with a CNN and use an LSTM network as the decoder, and the decoder is trained to maximize the log-likelihood estimation of the target captions. To mitigate the exposure bias problem, they adopt the schedule sampling approach [59], which selects between the ground-truth words and the machine-generated words according to the scheduled probability in training.

Only providing the global image feature is not sufficient, as the power of RNNs lies in its capability to model the contextual information between time steps, while the global image representation weakens the RNN’s memory of the visual information. Visual attention for image captioning was first being introduced by [60], which incorporates the spatial attention on convolutional features of images into the encoder-decoder framework through the *soft* and *hard* attention mechanisms. After that, some works have proposed to encode more discriminative visual information into the captioning model. They leverage visual attributes of the image to enhance the visual information using some weakly supervised approach. In [24, 61], they incorporate high-level attributes into the encoder-decoder framework and achieve large improvements. Both of [24] and [21] treat the attribute detection problem as multi-instance learning (MIL) problem and train a corresponding CNN by minimizing the element-wise logistic loss function. Liu *et al.* [62] use R-FCN [63] to detect the visual attributes and adopt a sequential attention mechanism to translate the attributes to a word sequence.

Although models like LSTM networks have memory cells that aim to memorize history information for long-term, it is still limited to several time steps because long-term information is gradually diluted at every time step [27]. Besides, vanilla RNN-based image captioning models recursively accumulate history information without explicitly modeling the hierarchical structure of word sequences, which clearly have a bottom-up structure [26]. In contrast to previous models that predict the next word based on one previous word and hidden state, our work in Chapter 3 proposes an image captioning model with language CNN to explore both hierarchical and temporal information of sequence for image caption generation.

2.2.2 Reinforcement Learning-Based Methods

Although the maximum likelihood estimation (MLE) can effectively train the encoder-decoder based image captioning models [18, 60, 62], models trained with MLE are still face the loss-evaluation mismatch problem [31, 64]. Specifically, loss-evaluation mismatch happens when a language model is trained with cross-entropy loss but evaluates the generated captions with the sentence-level evaluation metrics, *e.g.*, BLEU [65], CIDEr [66], *etc.*, which are non-differentiable and cannot be directly used as training loss.

Several attempts have been made to use Reinforcement Learning (RL) to address the discrepancy between the training and the testing objectives for image captioning [28]. The first work of training RNN-based sequence model with policy gradient was proposed by [64], in which a REINFORCE-based approach was used to calculate the sentence-level reward and a Monte-Carlo technique was employed for training. Similarly, Ranzato *et al.* [67] estimate the action value by averaging three roll-out sequences, which is the same as [68]. Instead of using the sentence-level reward in training, Bahdanau *et al.* [69] use the token-level reward in temporal difference training for sequence generation. Recently, the *self-critical* learning approach proposed by Rennie *et al.* [28] utilize an improved REINFORCE algorithm with a reward obtained by the current model against the baseline, *i.e.*, the inference algorithm.

All the above researches on image captioning mainly focus on the one-stage model. However, it is extremely hard for them to generate rich fine-grained descriptions. This is because rich descriptions require high-complexity models, where the problem of vanishing gradients often occurs, considering the back-propagated gradients diminish in strength as they propagate through many layers of a complex network. In Chapter 4, we propose a coarse-to-fine multi-stage prediction framework for

image captioning, composed of multiple decoders, each of which operates on the output of the previous stage, producing increasingly refined image descriptions.

2.2.3 Adversarial Learning-Based Methods

Instead of training the model with the handcrafted loss, some researchers applied the adversarial training for image captioning [68, 70], which uses adversarial training to encourage the dynamics of the RNNs to be the same as that of training conditioned on previous ground truth words.

Generative Adversarial Nets (GANs) [71] is first proposed to generate realistic images and has already been used successfully in computer vision tasks [30, 72, 73]. For image captioning, the generator learns the loss value provided by the discriminator. The discriminator can discriminate between generator-generated samples and real data samples, which allows the GAN-based image captioning methods to generate a diverse set of image captions in contrast to conventional encoder-decoder based methods.

Unlike the deterministic continuous mapping from random vectors to the image domain, the image captioning task is a discrete domain generation issue. Therefore, such operations are non-differentiable, making it difficult to apply back-propagation directly. Yu *et al.* [68] bypass the generator differentiation problem by directly performing the gradient policy update. In [70], they explore the use of conditional GANs [74] in generating image descriptions. Shetty *et al.* [75] explore the use of adversarial training in combination with an approximate Gumbel sampler [76, 77] in generating image descriptions. Although those GAN-based methods can generate diverse image descriptions, they sacrifice the fidelity and achieve much lower sentence-level scores compared with other image captioning algorithms.

2.3 Unpaired Image Captioning

Despite the impressive progress achieved by the paired image captioning, one of the key limitations is the availability of large paired image-caption datasets. Current image captioning task has so far been explored mainly in English. For some language, large scale image-caption paired data might not be available. Therefore, how to relieve the dependency on the paired image-caption datasets is becoming increasingly important.

Recently, there have been efforts to learn the encoder-decoder framework in an unsupervised fashion, such as unpaired sentence-to-sentence [29, 78] and image-to-image [30, 79] translations. However, only a few studies try to relax the reliance on paired image-caption training data for image captioning [12, 14, 80]. Arguably, compared to unpaired machine translation [29] and image translation [30], unpaired image-to-sentence translation is more challenging due to the significantly different characteristics between visual and textual modalities.

The first work in this direction is our pivot language-based image captioning model in Chapter 6. In our work, we assume that there is a second language called pivot language for which we have separate image-pivot and pivot-target paired datasets. Then, the image-to-target model in the pivot-based setting can be decomposed into two sub-models by treating the pivot sentence as a latent variable. However, the image-to-pivot and pivot-to-target models are quite different in terms of vocabulary and parameter space because they are trained on two possibly unrelated datasets. To overcome the limitations of the pivot-based caption generation, we propose to reduce the discrepancy between the image-to-pivot and pivot-to-target models, and to train them jointly in an end-to-end fashion so that they learn better models by interacting with each other during training.

After that, Fang *et al.* [80] move a step further, where they conduct unsupervised image captioning without relying on any labeled image-caption pairs. Their method

uses a sentence discriminator along with a visual concept detector to connect the image and the text modalities through adversarial training.

While impressive achievements have been made for the unpaired image captioning problem, this challenging task is far from mature. Inspired by the powerful semantic representation ability of scene graph [14], in Chapter 7, we adopt scene graphs as the intermediate representation of the image and the sentence, and map the scene graphs in their feature space through cycle-consistent adversarial training.

2.4 Cross-Modal Retrieval

Inspired by the success of deep neural networks [5], Frome *et al.* [81] propose a cross-modal feature embedding framework that uses CNN and Skip-Gram [82] to extract cross-modal feature representations, and then associate them with a structured objective in which the distance between the matched image-caption pair is smaller than that between the mismatched pair. A similar framework is proposed by Kiros *et al.* [83], in which a GRU is used as the sentence encoder. They also map the images and sentences to a common space and adopt the rank loss to penalize the model by averaging the individual violations across the negatives. Vendrov *et al.* [84] introduce an improved objective, which can preserve the partial order structure of a visual-semantic hierarchy. Klein *et al.* [85] adopt a similar objective and employ Fisher Vectors (FV) [86] as a pooling strategy of word embeddings for caption representation. In [87], they sell the visual word embedding idea and propose a joint image-caption embedding model for image captioning. However, their model is based on cartoon-like images, which is difficult to be applied to real images. Considering the strong ability of GANs in learning discriminative representation, Peng *et al.* [88] explore inter-modality and intra-modality with a cross-modal GAN. Recently, Faghri *et al.* [89] improve Kiros’s work by replacing the sum violations across the negative samples with the hardest negative samples.

Although the feature representations in the learned common representation space have been successfully used to describe high-level semantic concepts of multi-modal data, they are not sufficient to retrieve images with detailed local similarity (*e.g.*, spatial layout) or sentences with word-level similarity.

Several works have explored the alignment of visual objects and textual words [16, 90–92]. Karpathy *et al.* [90] use local alignment to embed the fragments of images and the sentences into a common space. Plummer *et al.* [16] go a step further and use all pairwise instances for similarity measurement. Jiang *et al.* [91] learn a multi-modal embedding by optimizing pairwise ranking, while enhancing both local alignment and global alignment. In [93], they introduce the context-modulated attention mechanism into the cross-modal embedding. Their attention scheme can selectively attend to pairwise instances of image and sentence, and then dynamically aggregate the measured similarity to obtain a global similarity between image and text. Instead of embedding the sentence with chain-structured RNNs, the recent work of Niu *et al.* [92] adopt a tree-structured LSTM to learn the hierarchical relations between sentences and images, and between phrases and visual objects.

Although the alignment of visual objects and textual words can enhance the local similarity between images and texts, however, in order to align visual objects with textual words, a sufficient amount of annotations need to be acquired as well, which induces expensive human annotations. Unlike existing image-text retrieval approaches that embed image-text pairs as single feature vectors in a common representational space, our method in Chapter 5 incorporates the image-to-text and the text-to-image generative models into the conventional textual-visual feature embedding, which is able to learn concrete grounded representations that capture the detailed similarity between the two modalities.

Chapter 3

An Empirical Study of Language CNN for Image Captioning

This chapter, together with Chapter 4, aims to tackle the problem of paired image captioning. Two aspects are studied: long-range dependencies of history words, and fine-grained descriptions. This chapter focuses on the first aspect.

3.1 Introduction

The recent surge of research interest in image caption generation task is due to the advances in machine translation [94] and large datasets [15, 16]. Most image caption models follow the encoder-decoder pipeline [20, 25, 95]. The encoder-decoder framework is recently introduced for sequence-to-sequence learning based on RNNs and LSTM networks. Both RNNs and LSTM networks can be sequence learners. However, due to the vanishing gradient problem, RNNs can only remember previously status for a few time steps. LSTM network is a special type of RNN architecture designed to solve the vanishing gradient problem in RNNs [18, 23, 54]. It introduces a new component called memory cell. Each memory cell is composed

of three gates and a neuron with a self-recurrent connection. These gates allow memory cells to keep and access information over a long period of time and make it capable of learning long-term dependencies.

Although models like LSTM networks have memory cells that aim to memorize history information for long-term, it is still limited to several time steps because long-term information is gradually diluted at every time step [27]. Besides, vanilla RNN-based image captioning models recursively accumulate history information without explicitly modeling the hierarchical structure of word sequences, which clearly have a bottom-up structure [26].

To better model the hierarchical structure and long-term dependencies of word sequences, in this work, we adopt a language CNN which applies temporal convolution to extract features of sequences. Such a method is inspired by works in NLP, which have shown CNN is very powerful for text representation [96, 97]. Unlike the vanilla CNN architecture, we drop the pooling operation to keep the relevant information for word representation and investigate the optimum convolutional filters by experiments. However, only using Language CNN fails to model the dynamic temporal behavior. Hence, we still need to combine Language CNN with recurrent networks (*e.g.*, RNN, or LSTM). Our study shows that adding Language CNN to a recurrent network helps model sequences consistently and effectively.

To summarize, our primary contribution is to show a language CNN that is capable of capturing long-range dependencies of sequence for image caption generation. Combining language CNN with RNNs can produce improved results. Our model yields comparable performance with the state-of-the-art approaches on Flickr30k [16] and MSCOCO [15].

FIGURE 3.1: An overview of our framework. The input of our model is a query image. Our model estimates the probability distribution of the next word given previous words and image. It consists of four parts: a $\text{CNN}_{\mathcal{I}}$ for image feature extraction, a deep $\text{CNN}_{\mathcal{L}}$ for language modeling, a multimodal layer (M) that connects the $\text{CNN}_{\mathcal{I}}$ and $\text{CNN}_{\mathcal{L}}$, and a Recurrent Network (*e.g.*, RNN, LSTM, *etc.*) for word prediction. The weights are shared among all time frames.

3.2 Methodology

3.2.1 Framework

We study the effect of Language CNN by combining it with Recurrent Networks. Figure 3.2 shows a recursive framework. It consists of one deep CNN for image encoding, one CNN for sentence modeling, and a recurrent network for sequence prediction. In order to distinguish these two CNN networks, we name the first CNN for image feature extraction as $\text{CNN}_{\mathcal{I}}$, and the second CNN for language modeling as $\text{CNN}_{\mathcal{L}}$.

Given an image $\mathbf{I}$, we take the widely-used CNN architecture VGGNet (16-layer) [98] pre-trained on ImageNet [99] to extract image features $\mathbf{V} \in \mathbb{R}^K$. The $\text{CNN}_{\mathcal{L}}$ is designed to represent words and their hierarchical structure in word sequences. It takes a sequence of t generated words (each word is encoded as a one-hot representation) as input and generates a bottom-up representation of these words. The outputs of $\text{CNN}_{\mathcal{I}}$ and $\text{CNN}_{\mathcal{L}}$ will be fed into a multimodal fusion layer, and use recurrent network $f_{\text{recurrent}}(\cdot)$ to predict the next word. The following equations

show the main working flow of our model:

$$\mathbf{V} = \text{CNN}_{\mathcal{I}}(\mathbf{I}) \quad (3.1)$$

$$\mathbf{y}^{[t]} = \text{CNN}_{\mathcal{L}}(\mathbf{S}^{[0]}, \mathbf{S}^{[1]}, \dots, \mathbf{S}^{[t-1]}) \quad (3.2)$$

$$\mathbf{m}^{[t]} = f_{\text{multimodal}}(\mathbf{y}^{[t]}, \mathbf{V}) \quad (3.3)$$

$$\mathbf{r}^{[t]} = f_{\text{recurrent}}(\mathbf{r}^{[t-1]}, \mathbf{x}^{[t-1]}, \mathbf{m}^{[t]}) \quad (3.4)$$

$$\mathbf{S}^{[t]} \sim \arg \max_{\mathcal{S}} \text{Softmax}(\mathbf{W}_o \mathbf{r}^{[t]} + \mathbf{b}_o) \quad (3.5)$$

where $t \in [0, N - 1]$ is the time step, $\mathbf{y}^{[t]}$ is the output vector of $\text{CNN}_{\mathcal{L}}$, $\mathbf{r}^{[t]}$ is the activation output of recurrent network, $\mathbf{S}^{[t]}$ is the t -th word drawn from the dictionary $\mathcal{S}$ according to the maximum Softmax probability controlled by $\mathbf{r}^{[t]}$, $\mathbf{W}_o$ and $\mathbf{b}_o$ are weights and biases used for calculating a distribution over words. Eq. (3.2), Eq. (3.3), Eq. (3.4) and Eq. (3.5) are recursively applied, the design of each function is discussed below.

3.2.1.1 $\text{CNN}_{\mathcal{L}}$ Layer

Current sequence modeling tasks have been dominated by LSTM networks [25, 94, 100, 101]. However, LSTM networks cannot explicitly model the hierarchical representation of words. Even with multi-layer LSTM networks, such hierarchical structure is still hard to be captured due to the more complex model and the higher risk of over-fitting.

Inspired by the recent success of CNNs in computer vision [5, 102, 103]. We adopt a language CNN with hierarchical structure to capture the long-range dependencies of the input words, called $\text{CNN}_{\mathcal{L}}$. The first layer of $\text{CNN}_{\mathcal{L}}$ is a word embedding layer. It embeds the one-hot word encoding from the dictionary into a word representation through a lookup table. Suppose we have t input words $\mathbf{S} = \{\mathbf{S}^{[0]}, \mathbf{S}^{[1]}, \dots, \mathbf{S}^{[t-1]}\}$, and $\mathbf{S}^{[i]}$ is the one-of- V (one-hot) encoding, with V as

the size of the vocabulary. We first map each word $S^{[t]}$ in the sentence into a K -dimensional vector $\mathbf{x}^{[t]} = \mathbf{W}_e S^{[t]}$, where $\mathbf{W}_e \in \mathbb{R}^{K \times V}$ is a word embedding matrix (to be learned). Next, those embeddings are concatenated transversely to produce a matrix as follows:

$$\mathbf{x} = [\mathbf{x}^{[0]}; \mathbf{x}^{[1]}; \dots; \mathbf{x}^{[t-1]}]^T, \mathbf{x} \in \mathbb{R}^{t \times K} \quad (3.6)$$

The concatenated matrix $\mathbf{x}$ is fed to the convolutional layer. Just like a normal CNN, $\text{CNN}_{\mathcal{L}}$ has a fixed architecture with predefined maximum number of input words (denoted as $L_{\mathcal{L}}$). Unlike the toy example in Figure 3.2, in practice we use a larger and deeper $\text{CNN}_{\mathcal{L}}$ with $L_{\mathcal{L}} = 16$.

We use temporal convolution [104] to model the sentence. Given an input feature map $\mathbf{y}^{(\ell-1)} \in \mathbb{R}^{M_{\ell-1} \times K}$ of Layer- $\ell - 1$, the output feature map $\mathbf{y}^{(\ell)} \in \mathbb{R}^{M_{\ell} \times K}$ of the temporal convolution layer- ℓ will be:

$$y_i^{(\ell)}(\mathbf{x}) = \sigma(\mathbf{w}_L^{(\ell)} \mathbf{y}_i^{(\ell-1)} + b_L^{(\ell)}) \quad (3.7)$$

here $y_i^{(\ell)}(\mathbf{x})$ gives the output of feature map for location i in Layer- ℓ , $\mathbf{w}_L^{(\ell)}$ denotes the parameters on Layer- ℓ , $\sigma(\cdot)$ is the activation function, *e.g.*, Sigmoid, or ReLU. The input feature map $\mathbf{y}_i^{(\ell-1)}$ is the segment of Layer- $\ell - 1$ for the convolution at location i , while $\mathbf{y}^{(0)}$ is the concatenation of t word embeddings from the sequence input $S^{[0:t-1]}$. The definition of $\mathbf{y}^{(0)}$ is as follows:

$$\mathbf{y}^{(0)} \stackrel{\text{def}}{=} \begin{cases} [\mathbf{x}^{[t-L_{\mathcal{L}}]}; \dots; \mathbf{x}^{[t-1]}], & \text{if } t \geq L_{\mathcal{L}} \\ [\mathbf{x}^{[0]}; \dots; \mathbf{x}^{[t-1]}; \tilde{\mathbf{x}}^{[t]}; \dots; \tilde{\mathbf{x}}^{[L_{\mathcal{L}}-1]}] & \text{otherwise} \end{cases} \quad (3.8)$$

Specially, when $t \geq L_{\mathcal{L}}$, the input sentence will be truncated, we only use $L_{\mathcal{L}}$ words before the current time step t . When $t < L_{\mathcal{L}}$, the input sentence will be padded with $\tilde{\mathbf{x}}^{[i]}$. Note that if $t = 0$, $\tilde{\mathbf{x}}^{[i]}$ are the image features $\mathbf{V}$, otherwise $\tilde{\mathbf{x}}^{[i]}$ are the

zero vectors that have the same dimension as $x^{[i]}$.

Previous CNNs, including those adopted for NLP tasks [96, 105], take a classic *convolution-pooling* strategy, which uses max-pooling to pick the highest response feature across time. This strategy works well for tasks like text classification [96] and matching [105], but is undesirable for modeling the composition functionality, because it ignores temporal information of sequence. In our network, we discard the pooling operations. We consider words as the smallest linguistic unit and apply a straightforward stack of convolution layers on top of each other. In practice, we found that deeper $\text{CNN}_{\mathcal{L}}$ works better than shallow $\text{CNN}_{\mathcal{L}}$, which is consistent with the tradition of CNNs in computer vision [5], where using very deep CNNs is key to having better feature representation.

The output features of the final convolution layer are fed into a fully connected layer that projects the extracted words features into a low-dimensional representation. Next, the projected features will be fed to a highway connection [106], which controls flows of information in the layer and improves the gradient flow. The final output of the highway connection is a K -dimensional vector $y^{[t]}$.

FIGURE 3.2: The architecture of language CNN for sentence modeling. Here “/” stands for a zero padding. The $\text{CNN}_{\mathcal{L}}$ builds a hierarchical representation of history words that contains useful information for the next word prediction.

3.2.1.2 Multimodal Fusion Layer

Next, we add a multimodal fusion layer after $\text{CNN}_{\mathcal{L}}$, which fuses words representation and image features. This layer has two inputs: the bottom-up words representation $y^{[t]}$ extracted from the $\text{CNN}_{\mathcal{L}}$ and the image representation V extracted from $\text{CNN}_{\mathcal{I}}$. We map these two inputs to the same multimodal feature space and combine them together to obtain the activation of the multimodal feature:

$$m^{[t]} = f_{\text{multimodal}}(y^{[t]}, V) \quad (3.9)$$

$$= \sigma(f_y(y^{[t]}; \mathbf{W}_Y, \mathbf{b}_Y) + g_v(V; \mathbf{W}_V, \mathbf{b}_V)) \quad (3.10)$$

where “+” denotes element-wise addition, $f_y(\cdot)$ and $g_v(\cdot)$ are linear mapping functions, $m^{[t]}$ is the multimodal layer output feature vector. $\sigma(\cdot)$ is the activation function, here we use the scaled tanh function [42] which leads to a faster training process than the basic tanh function.

3.2.1.3 Recurrent Networks

Our $\text{CNN}_{\mathcal{L}}$ may miss the important temporal information because it extracts the holistic features for the whole sequence of words. To overcome this limitation, we combine it with recurrent networks. In our model, the transition equations of the recurrent network can be formulated as:

$$r^{[t]} = f_{\text{recurrent}}(r^{[t-1]}, x^{[t-1]}, m^{[t]}), \quad (3.11)$$

$$S^{[t]} \sim \arg \max_S \text{Softmax}(\mathbf{W}_o r^{[t]} + \mathbf{b}_o) \quad (3.12)$$

where $r^{[t]}$ denotes the recurrent state, $x^{[t-1]} = \mathbf{W}_e S^{[t-1]}$ is the previous word embedding, $m^{[t]}$ is the multimodal fusion output, and $f_{\text{recurrent}}(\cdot)$ is the transition function of recurrent network. $\text{Softmax}(r^{[t]})$ is the probability of word $S^{[t]}$ given by a Softmax layer, and $S^{[t]}$ is the t -th decoded word. In our study, we combine

our Language CNN with four types of recurrent networks: Simple RNN, LSTM network, GRU, and Recurrent Highway Network (RHN) [107].

Traditionally, the simple RNNs update the recurrent state $\mathbf{r}^{[t]}$ of Eq. (3.11) as follows:

$$\mathbf{r}^{[t]} = \tanh(\mathbf{W}_r \mathbf{r}^{[t-1]} + \mathbf{W}_z \mathbf{z}^{[t]} + \mathbf{b}) \quad (3.13)$$

where $\mathbf{z}^{[t]}$ is the input. However, this type of simple RNN is hard to deal with long-term dependencies [52]. As the vanishing gradient will make gradients in directions that short-term dependencies are large, while the gradients in directions that correspond to long-term dependencies are small.

LSTM network extends the simple RNN with the gating mechanism (*input* gate, *forget* gate, and *output* gate) to control information flow and an *memory cell* to store the history information, thus it can better model the long-term dependencies than simple RNNs.

GRU is an architecture similar to the LSTM, but it has a simplified structure. GRU does not have a separate *memory cell* and exposes its hidden state $\mathbf{r}^{[t]}$ without any control. Thus, it is computationally more efficient and outperforms the LSTM network on many tasks due to its simple structure.

FIGURE 3.3: Illustration of (a) Simple RNN, and (b) RHN. $\tilde{\mathbf{r}}$ is the hidden state, $\mathbf{r}$ is the new hidden state.

Besides, we also consider a fourth type of recurrent network: RHN (see Figure 3.3). RHN introduces the highway connection to simple RNN. It has directly gated connections between previous state $\mathbf{r}^{[t-1]}$ and current input $\mathbf{z}^{[t]}$ to modulate the flow of information. The transition equations of RHN can be formulated as follows:

$$\begin{pmatrix} \mathbf{t}^{[t]} \\ \mathbf{c}^{[t]} \\ \mathbf{h}^{[t]} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(\mathbf{M} \begin{pmatrix} \mathbf{r}^{[t-1]} \\ \mathbf{z}^{[t]} \end{pmatrix} \right) \quad (3.14)$$

$$\mathbf{r}^{[t]} = \mathbf{h}^{[t]} \odot \mathbf{t}^{[t]} + \mathbf{c}^{[t]} \odot \mathbf{r}^{[t-1]} \quad (3.15)$$

where $\mathbf{c}^{[t]}$ is the *carry* gate, $\mathbf{t}^{[t]}$ is the *transform* gate, $\mathbf{h}^{[t]}$ denotes the modulated input, $\mathbf{M} : \mathbb{R}^{2K+d} \rightarrow \mathbb{R}^{3d}$ is an affine transformation. $\mathbf{z}^{[t]} \in \mathbb{R}^{2K}$ denotes the concatenation of two vectors: $\mathbf{m}^t$ and $\mathbf{x}^{t-1}$. According to Eq. (3.3) and Eq. (3.2), $\mathbf{z}^{[t]}$ can be expressed as follows:

$$\mathbf{z}^{[t]} = [f_{\text{multimodal}}(\text{CNN}_{\mathcal{L}}(\mathbf{x}^{[0,\dots,t-1]}), \mathbf{V}); \mathbf{x}^{[t-1]}] \quad (3.16)$$

Like GRU, RHN does not have *output* gate to control the exposure of the recurrent state $\mathbf{r}^{[t]}$, but exposes the whole state each time. The RHN, however, does not have *reset* gate to drop information that is irrelevant in the future. As our $\text{CNN}_{\mathcal{L}}$ can extract the relevant information from the sequence of history words at each time step, to some extent, the $\text{CNN}_{\mathcal{L}}$ allows the model to add information that is useful in making a prediction.

3.2.2 Learning

During training, given the ground truth words $\mathbf{S}$ and corresponding image $\mathbf{I}$, the loss function for a single training instance $(\mathbf{S}, \mathbf{I})$ is defined as a sum of the negative

log-likelihood of the words. The loss can be written as:

$$\mathcal{L}(\mathbf{S}, \mathbf{I}) = - \sum_{t=0}^{N-1} \log P(S^{[t]} | S^{[0]}, \dots, S^{[t-1]}, \mathbf{I}) \quad (3.17)$$

where N is the sequence length, and $S^{[t]}$ denotes a word in the sentence $\mathbf{S}$.

The training objective is to minimize the cost function, which is equivalent to maximizing the probability of the ground truth context words given the image by using the following formulation:

$$\arg \max_{\theta} \sum_{t=0}^{N-1} \log P(S^{[t]} | S^{[0:t-1]}, \mathbf{I}) \quad (3.18)$$

where θ are the parameters of our model. $P(S^{[t]} | S^{[0:t-1]}, \mathbf{I})$ corresponds to the activation of the Softmax layer.

3.3 Experiments

3.3.1 Datasets and Evaluation Metrics

We perform experiments on two popular datasets that are used for image caption generation, MSCOCO and Flickr30k. These two datasets contain 123,000 and 31,000 images respectively, and each image has five reference captions. For MSCOCO, we reserve 5,000 images for validation and 5,000 images for testing. For Flickr30k, we use 29,000 images for training, 1,000 images for validation, and 1,000 images for testing.

We choose four metrics for evaluating the quality of the generated sentences:

BLEU- n [65] is a precision-based metric. It measures how many words are shared by the generated captions and ground truth captions.

METEOR [108] is based on the explicit word to word matches between generated

captions and ground-truth captions.

CIDEr [66] is a metric developed specifically for evaluating image captions. It measures consensus in image caption by performing a Term Frequency-Inverse Document Frequency weighting for each n -gram.

SPICE [109] is a more recent metric which has been shown to correlate better with the human judgment of semantic quality than previous metrics.

3.3.2 Models

To gain insight into the effectiveness of $\text{CNN}_{\mathcal{L}}$, we compare $\text{CNN}_{\mathcal{L}}$ -based models with methods using recurrent network only. For a fair comparison, the output dimensions of all gates are fixed to 512.

Recurrent Network-based Models. We implement Recurrent Network-based Models based on the framework proposed by Vinyals *et al.* [18], it takes an image as input and predicts words with one-layer Recurrent Network. Here we use the publicly available implementation NeuralTalk2¹. We evaluate four baseline models: **Simple RNN**, **RHN**, **LSTM**, and **GRU**.

$\text{CNN}_{\mathcal{L}}$ -based Models. As can be seen in Figure 3.1. The $\text{CNN}_{\mathcal{L}}$ -based models employ a $\text{CNN}_{\mathcal{L}}$ to obtain the bottom-up representation of a sequence of words and cooperates with the Recurrent Network to predict the next word. Image features and words representation learned from $\text{CNN}_{\mathcal{I}}$ and $\text{CNN}_{\mathcal{L}}$ respectively are fused with the multimodal function. We implement four $\text{CNN}_{\mathcal{L}}$ -based models: **$\text{CNN}_{\mathcal{L}}$ +Simple RNN**, **$\text{CNN}_{\mathcal{L}}$ +RHN**, **$\text{CNN}_{\mathcal{L}}$ +LSTM**, and **$\text{CNN}_{\mathcal{L}}$ +GRU**.

We set a fixed beam search size k for all models. To analyze the effect of the beam search size k in the testing stage, we compare the performances of the $\text{CNN}_{\mathcal{L}}$ +RHN model and $\text{CNN}_{\mathcal{L}}$ +LSTM model with different beam sizes. We find beam size $k = 2$ achieves better performance. Hence, we set the beam size k as 2 in our tests.

¹<https://github.com/karpathy/neuraltalk2>

3.3.3 Implementation Details

In the following experiments, we use the 16-layer VGGNet [98] model to compute CNN features and map the last fully-connected layer’s output features to an embedding space via a linear transformation.

3.3.3.1 Data Preprocessing

As for the preprocessing of captions, we transform all letters in the captions to lowercase and remove all the non-alphabetic characters. Words that occur less than five times are replaced with an unknown token $\langle \text{UNK} \rangle$. We truncate all the captions longer than 16 tokens and set the maximum number of input words for $\text{CNN}_{\mathcal{L}}$ to be 16.

3.3.3.2 Training Details

In the training process, each image $\mathbf{I}$ has 5 corresponding annotations. We first extract the image features $\mathbf{V}$ with $\text{CNN}_{\mathcal{I}}$. The image features $\mathbf{V}$ are used in each time step. We map each word representation $\mathbf{S}^{[t]}$ with: $\mathbf{x}^{[t]} = \mathbf{W}_e \mathbf{S}^{[t]}, t \in [0, N-1]$. After that, our network is trained to predict the words after it has seen the image and the preceding words. Please note that we denote by $\mathbf{S}^{[0]}$ a special $\langle \text{START} \rangle$ token and by $\mathbf{S}^{[N-1]}$ a special $\langle \text{END} \rangle$ token which designates the start and end of the sentence.

For Flickr30K [16] and MSCOCO [15], we set the dimensionality of the image features and word embeddings as 512. All the models are trained with Adam [110], which is a stochastic gradient descent method that computes an adaptive learning rate for each parameter. The learning rate is initialized with $2\text{e-}4$ for Flickr30K and $4\text{e-}4$ for MSCOCO, and a restart technique mentioned in [111] is adopted to improve the convergence of training. Dropout and early stopping are used to avoid

overfitting. All weights are randomly initialized except for the CNN weights. More specifically, we fine-tune the VGGNet when the validation loss stops decreasing. The termination of training is determined by evaluating the CIDEr [66] score for the validation split after each training epoch.

3.3.3.3 Testing

During testing, the previous output $S^{[t-1]}$ is used as input in lieu of $S^{[t]}$. The sentence generation process is straightforward. Our model starts from the $\langle \text{START} \rangle$ token and calculates the probability distribution of the next word : $P(S^{[t]}|S^{[0:t-1]}, \mathbf{I})$. Here we use Beam Search technology proposed in [23], which is a fast and efficient decoding method for recurrent network models.

3.3.4 Quantitative Results

We compare those models mentioned in Section 3.3.2 on MSCOCO and Flickr30K datasets to evaluate the effects of $\text{CNN}_{\mathcal{L}}$ and report the results in Table 3.1 and Table 3.2.

TABLE 3.1: Performance comparison on MSCOCO, where M is short for METEOR, and S is short for SPICE (Bold numbers are the best results).

	B@1	B@2	B@3	B@4	M	C	S
$\text{CNN}_{\mathcal{L}}$	54.6	39.9	27.5	18.4	17.8	56.8	12.2
Simple RNN	70.1	52.1	37.6	27.0	23.2	87.0	16.0
$\text{CNN}_{\mathcal{L}}$ +RNN	72.2	55.0	40.7	29.5	24.5	95.2	17.6
RHN	70.5	52.7	37.8	27.0	24.0	90.6	17.2
$\text{CNN}_{\mathcal{L}}$ +RHN	72.3	55.3	41.3	30.6	25.2	98.9	18.3
LSTM	70.8	53.6	39.5	29.2	24.5	92.6	17.1
$\text{CNN}_{\mathcal{L}}$ +LSTM	72.1	54.6	40.9	30.4	25.1	99.1	18.0
GRU	71.6	54.1	39.7	28.9	24.3	93.3	17.2
$\text{CNN}_{\mathcal{L}}$ +GRU	72.6	55.4	41.1	30.3	24.6	96.1	17.6

TABLE 3.2: Performance comparison on Flickr30k (Bold numbers are the best results).

	B@1	B@2	B@3	B@4	M	C	S
Simple RNN	60.5	41.3	28.0	19.1	17.1	32.5	10.5
CNN _L +RNN	71.3	53.8	39.6	28.7	22.6	65.4	15.6
RHN	62.1	43.1	29.4	20.0	17.7	38.4	11.4
CNN _L +RHN	73.8	56.3	41.9	30.7	21.6	61.8	15.0
LSTM	60.9	41.8	28.3	19.3	17.6	35.0	11.1
CNN _L +LSTM	64.5	45.8	32.2	22.4	19.0	45.0	12.5
GRU	61.4	42.5	29.1	20.0	18.1	39.5	11.4
CNN _L +GRU	71.4	54.0	39.5	28.2	21.1	57.9	14.5

3.3.4.1 Results Using CNN_L on MSCOCO

Table 3.1 shows the generation performance on MSCOCO. By combining CNN_L, our methods clearly outperform the recurrent network counterpart in all metrics.

Among these models, CNN_L+RHN achieves the best performances in terms of BLEU-(3,4), METEOR, and SPICE metrics, CNN_L+LSTM achieves the best performance in CIDEr metric (99.1), and CNN_L+GRU achieves the best performance in BLEU-(1,2) metrics. We also found that those predicted captions of CNN_L only are short, but contain primary attributes, *e.g.*, CNN_L model generates: “*a person on a wave*”, while CNN_L+RNN provides: “*a young man surfing a wave*”. This shows that the temporal recurrence of RNN is crucial for modeling the short-term contextual information across words in the sentence.

Note that the CNN_L+RNN model achieves better performance than simple RNN model and outperforms the LSTM model. As mentioned in Section 3.2.1.3, LSTM networks model the word dependencies with multi-gates and the internal memory cell. However, our CNN_L+RNN without memory cell works better than the LSTM model. We think the reason is that our language CNN takes all history words as input and explicitly model the long-term dependencies of history words; this could be regarded as an external “*memory cell*”. Thus, the CNN_L’s ability to model

long-term dependencies can be taken as enhancement of simple RNNs, which can solve the difficulty of learning long-term dependencies.

3.3.4.2 Results Using $\text{CNN}_{\mathcal{L}}$ on Flickr30K

We also evaluate the effectiveness of language CNN on the smaller dataset Flickr30K. The results in Table 3.2 clearly indicate the advantage of exploiting the language CNN to model the long-term dependencies of words for image captioning. Among all models, $\text{CNN}_{\mathcal{L}}$ +RHN achieves the best performances in BLEU-(1,2,3,4) metrics, and $\text{CNN}_{\mathcal{L}}$ +RNN achieves the best performances in METEOR, CIDEr, and SPICE metrics.

As for the low results (without $\text{CNN}_{\mathcal{L}}$) on Flickr30k, we think that it is due to lack of enough training data to avoid overfitting. In contrast, our $\text{CNN}_{\mathcal{L}}$ can help learn better word embedding and history words representation for word prediction, and it is much easier to be trained compared with LSTM due to its simplicity and efficiency. Note that the performance of LSTM and $\text{CNN}_{\mathcal{L}}$ +LSTM models are lower than RHN/GRU and $\text{CNN}_{\mathcal{L}}$ +RHN/GRU. This illustrates that the LSTM networks are easily overfitting on this smaller dataset.

3.3.4.3 Comparison with State-of-the-art Methods

To empirically verify the merit of our models, we compare our methods with other state-of-the-art methods.

Performance on MSCOCO. The right-hand side of Table 3.3 shows the results of different models on MSCOCO dataset. $\text{CNN}_{\mathcal{L}}$ -based models perform better than most image captioning models. The only two methods with better performance (for some metrics) than ours are Attributes-CNN+RNN [21] and Google NICv2 [18].

FIGURE 3.4: Qualitative results for images on MSCOCO. Ground-truth annotations (under each dashed line) and the generated annotations are shown for each image.

FIGURE 3.5: Some failure descriptions for images on MSCOCO, Ground-truth descriptions are under each dashed line.

However, *wu et al.* [21] employ an attribute prediction layer, which requires determining an extra attribute vocabulary. While we generate the image descriptions only based on the image features. Google NICv2 [18] is based on Google NIC [112]; the results of Google NICv2 are achieved by model ensembling. Despite all this, the CIDEr score of our $CNN_{\mathcal{L}}+LSTM$ model can still achieve 99.1%, which is comparable to their best performance even with a single VGG-16 model.

Performance on Flickr30K. The results on Flickr30K are reported in the left-hand side of Table 3.3. Interestingly, $\text{CNN}_{\mathcal{L}}+\text{RHN}$ performs the best on this smaller dataset and even outperforms the latest state-of-the-art model (Attributes-CNN+RNN [21]).

Obviously, there is a significant performance gap between $\text{CNN}_{\mathcal{L}}+\text{RNN}/\text{RHN}/\text{GRU}$ and $\text{RNN}/\text{RHN}/\text{GRU}/\text{LSTM}$ models. This demonstrates the effectiveness of our language CNN on the one hand, and also shows that our $\text{CNN}_{\mathcal{L}}+\text{RNN}/\text{RHN}/\text{GRU}$ models are more robust and easier to train than LSTM networks when less training data is available.

3.3.5 Qualitative Results

Figure 3.4 shows some examples generated by our models. It is easy to see that all of these caption generation models can generate somewhat relevant sentences, while the $\text{CNN}_{\mathcal{L}}$ -based models can predict more high-level words by jointly exploiting history words and image representations. Take the last image as an example, compared with the sentences generated by $\text{RNN}/\text{LSTM}/\text{GRU}$ model, “*a cat is looking at a dog in front of a window*” generated by $\text{CNN}_{\mathcal{L}}+\text{RNN}$ is more precise to describe their relationship in the image.

Besides, our $\text{CNN}_{\mathcal{L}}$ -based model can generate more descriptive sentences. For instance, for the first image, with the detected object “*cat*”, the generated sentence “*a black and white cat looking at itself in a mirror*” by $\text{CNN}_{\mathcal{L}}+\text{RHN}$ depicts the image content more comprehensively. The results demonstrate that our model with language CNN can generate more humanlike sentences by modeling the hierarchical structure and long-term information of words.

Figure 3.5 shows some failure samples of our $\text{CNN}_{\mathcal{L}}$ -based model. Although most of the generated captions are complete sentences. However, the biggest problem is that those predicted visual attributes are wrong. For example, “*bear*” in the

first image is detected as “*bird*”, and “*brown*” in the second image is detected as “*black and white*”. This will decrease the precision-based evaluation score (*e.g.*, BLEU- n). We can improve our model by further taking high-level attributes into account.

3.4 Summary

In this chapter, we present an image captioning model with language CNN to explore both hierarchical and temporal information of sequence for image caption generation. Experiments conducted on MSCOCO and Flickr30K image captioning datasets validate our proposal and analysis. Performance improvements are clearly observed when compared to other image captioning methods.

Chapter 4

Coarse-to-Fine Learning for Image Captioning

4.1 Introduction

The challenge of image captioning lies in designing a model that can effectively utilize the image information and generate more human-like rich image descriptions. Motivated by the recent advances in natural language processing, current image captioning approaches typically follow the encoding-decoding framework [64], which consists of a CNN based image encoder and a RNN based sentence decoder, with various variants for image captioning [19–21]. Most of these existing image captioning approaches are trained by maximizing the likelihood of each ground-truth word given the previous ground-truth words and the image with backpropagation.

There are three major problems in these existing image captioning methods. Firstly, it is extremely hard for them to generate rich fine-grained descriptions. This is because rich descriptions require high-complexity models, where the problem of vanishing gradients often occurs, considering the back-propagated gradients diminish

FIGURE 4.1: Illustration of our proposed coarse-to-fine framework. Our model consists of one image encoder (CNN) and a sequence of sentence decoders (attention-based LSTM networks), and it takes the image as input and refines the image descriptions from coarse to fine. Here we show the increasingly improved image descriptions in two stages (gray and dark gray).

in strength as they propagate through many layers of a complex network. Secondly, there is an exposure bias between the training and the testing [64, 115, 116]. Specifically, the sentence decoder is trained to predict a word given the previous ground-truth words, while at testing time, the caption generation is accomplished by greedy search or with beam search, which predicts the next word based on the previously generated words that is different from the training mode. Since the model has never been exposed to its own predictions, it will result in error accumulation at test time. To address the exposure bias problem, scheduled sampling [59], *i.e.*, randomly selecting between previous ground-truth words and previously generated words, has become the current dominant training procedure to fit RNNs based models. However, it can only mitigate the exposure bias but cannot largely solve it. Thirdly, there is a loss-evaluation mismatch [64]. Specifically, language models are usually trained to minimize the cross-entropy loss at each time-step, while at testing time, we evaluate the generated captions with the sentence-level evaluation metrics, *e.g.*, BLEU- n [65], CIDEr [66], SPICE [109], *etc.*,

which are non-differentiable and cannot be directly used as training loss.

In this work, considering the great challenge of generating rich image descriptions in one stage, we propose a coarse-to-fine multi-stage prediction framework. Our model consists of an image encoder and a sequence of sentence decoders that repeatedly generate image descriptions in finer details. However, directly composing such multi-stage decoders in an image captioning model faces the risk of the vanishing gradients problem. Motivated by the works on image recognition [117, 118], which show that supervising very deep networks at intermediate layers aids in learning, we also enforce intermediate supervisions for each stage. Furthermore, inspired by the recent image captioning work [28], which uses Reinforcement Learning to address the loss-evaluation mismatch problem and include the inference process as a baseline in training to address the exposure bias problem, we also design a similar RL-based training method but extend it from one-stage [28] to our multi-stage framework, where rewards are introduced at each stage as intermediate supervision. Particularly, we optimize our model with a RL-based approach, which utilizes the output of each intermediate decoder’s test-time inference algorithm as well as the output of its preceding decoder to normalize the rewards. In addition, to cope with our coarse-to-fine learning framework, we adopt a stacked attention model to extract more fine-grained visual attention information for word prediction at each stage. Figure 4.1 illustrates our proposed coarse-to-fine framework, which consists of three stacked LSTM networks. The first LSTM generates the coarse-scale image description, and the subsequent LSTM networks serve as the fine-scale decoders. At each stage in our model, attention weights, and hidden vector produced by the preceding stage are used as inputs, which are taken as the disambiguating cues to the subsequent stage. As a result, each stage of the decoder generates words with increasingly refined attention weights as well as words.

The main contributions of this work include: (a) a coarse-to-fine framework which increases the model complexity gradually with increasingly refined attention weights

for image captioning and (b) a reinforcement learning method that directly optimizes model with the normalized intermediate rewards. Experiments show the outstanding performance of our approach on MSCOCO [15].

4.2 Methodology

In this work, we consider the problem of learning to generate image description $\hat{\mathbf{Y}} = \{\hat{Y}_0, \dots, \hat{Y}_{T-1}\}$ for an image $\mathbf{I}$, where $\hat{Y}_t \in \mathcal{D}$ is the predicted word, $\mathcal{D}$ is the dictionary, and T denotes the sequence length. Our algorithm builds a coarse-to-fine model with the same target as those one-stage models, but with the additional intermediate layers between the output layer and the input layer. We first train the model by maximizing log-likelihood of each successive target word conditioned on the input image and the gold history of target words $\mathbf{Y} = \{Y_0, \dots, Y_{T-1}\}$, and then optimize the model with sentence-level evaluation metrics. We denote by $\hat{\mathbf{Y}}^i, i \in \{0, \dots, N_f\}$ the predicted word sequence of the i -th stage decoder, and N_f is the total number of fine stages. As a result, each intermediate sentence decoder predicts the increasingly refined image description, and the prediction of the last decoder is taken as the final image description. Note that we treat stage $i = 0$ as the coarse decoder, and stages $i \geq 1$ as the fine decoders.

4.2.1 Image Encoding

We first encode the given image $\mathbf{I}$ to the spatial image features $\mathbf{V} = \{V_0, \dots, V_{k \times k - 1}\}$, $V_i \in \mathbb{R}^{d_v}$ with CNN: $\mathbf{V} = \text{CNN}(\mathbf{I})$, where $k \times k$ is the number of regions, each feature channel V_i depicts a region of the image, and d_v is the dimension of the feature vector for each region. Specifically, we extract the image features from the final convolutional layer of CNN, and use spatial adaptive average pooling to resize the features to a fixed-size spatial representation of $k \times k \times d_v$.

4.2.2 Coarse-to-Fine Decoding

The overall coarse-to-fine sentence decoder consists of one coarse decoder and a sequence of attention-based fine decoders that repeatedly produce refined attention maps for the prediction of each word based on the cues from the preceding decoder. The first stage of our model is a coarse decoder, which predicts a coarse description from the global image feature. In the subsequent stages, each stage $i \in \{1, \dots, N_f\}$ is a fine decoder which predicts the improved image description based on image features and the outputs of the preceding stage. Particularly, we use the attention weights of the preceding stage to provide the following stage beliefs of regions for word prediction. More formally, we decode the image features in multiple stages, where the prediction $\hat{\mathbf{Y}}^i$ of each stage is a refinement of the prediction $\hat{\mathbf{Y}}^{i-1}$ of the previous stage.

Figure 4.2 illustrates the coarse-to-fine decoding architecture, where the top row contains one coarse decoder and two stacked attention-based fine decoders under the training mode, and the bottom row shows the fine decoders under its inference mode (greedy decoding) for computing rewards so as to incorporate intermediate supervisions. In the following, we will introduce the adopted coarse decoder, our proposed fine decoder, our proposed stacked attention model and our proposed RL-based process for incorporating intermediate supervisions.

4.2.2.1 Coarse Decoder.

We start by decoding in a coarse search space in the first stage ($i = 0$), where we learn a coarse decoder with an LSTM network, called $\text{LSTM}_{\text{coarse}}$. At each time step $t \in [0, T - 1]$, the input to $\text{LSTM}_{\text{coarse}}$ consists of the previous target word y_{t-1} , concatenated with the global image feature, and the previous hidden states.

The operation of the $\text{LSTM}_{\text{coarse}}$ can be described as:

$$o_t^0, h_t^0 = \text{LSTM}_{\text{coarse}}(h_{t-1}^0, i_t^0, y_{t-1}) \quad (4.1)$$

$$i_t^0 = [f(\mathbf{V}); h_{t-1}^{N_f}] \quad (4.2)$$

where h_{t-1}^0 and $h_{t-1}^{N_f}$ are the hidden states, o_t^0 is the cell output, $y_{t-1} = \mathbf{W}_e Y_{t-1}$ is the embedding of previous word Y_{t-1} . We obtain the global image feature $f(\mathbf{V})$ by taking a mean-pooling over the spatial image features as $\frac{1}{k \times k} \sum_{i=0}^{k \times k - 1} V_i$. The t -th decoded word $\hat{Y}_t^0$ of $\text{LSTM}_{\text{coarse}}$ is drawn from the dictionary $\mathcal{D}$ according to the softmax probability: $\hat{Y}_t^0 \sim \text{Softmax}(\mathbf{W}_o^0 o_t^0 + \mathbf{b}_o^0)$.

4.2.2.2 Fine Decoder.

In the subsequent stages, each fine decoder predicts the word $\hat{Y}_t^i$ based on the image features $\mathbf{V}$ again, and the attention weights $\boldsymbol{\alpha}_t^{i-1}$ and the hidden state h_t^{i-1} from the preceding LSTM. Each fine decoder consists of an $\text{LSTM}_{\text{fine}}^i$ network and an attention model. At each time step t , the input to $\text{LSTM}_{\text{fine}}^i$ consists of the attended image feature, the previous word embedding y_{t-1} , its previous hidden state h_{t-1}^i , and the updated hidden state h_t^{i-1} from the preceding LSTM. Note that when $t = 1$, h_t^0 is the hidden output of $\text{LSTM}_{\text{coarse}}$; otherwise h_t^{i-1} is the hidden output of the preceding $\text{LSTM}_{\text{fine}}^{i-1}$. Therefore, the updating procedure of $\text{LSTM}_{\text{fine}}^i$ can be written as:

$$o_t^i, h_t^i = \text{LSTM}_{\text{fine}}^i(h_{t-1}^i, i_t^i, y_{t-1}) \quad (4.3)$$

$$i_t^i = [g(\mathbf{V}, \boldsymbol{\alpha}_t^{i-1}, h_t^{i-1}); h_t^{i-1}] \quad (4.4)$$

where o_t^i is the cell output of $\text{LSTM}_{\text{fine}}^i$, and $g(\cdot)$ is the spatial attention function which feeds attended visual representations as the additional inputs to $\text{LSTM}_{\text{fine}}^i$ at each time step to emphasise the detailed visual information. During the inference,

FIGURE 4.2: Illustration of the proposed coarse-to-fine decoding using intermediate supervision (reward) after each stage. The top row (gray) contains one coarse decoder (left) and two visual attention-based fine decoders under the training mode. The bottom row shows the fine decoders under its inference mode (greedy decoding) for computing rewards.

the final output word $\hat{Y}_t$ is drawn from $\mathcal{D}$ according to the softmax probability:

$$\hat{Y}_t \sim \text{Softmax}(\mathbf{W}_o^{N_f} o_t^{N_f} + \mathbf{b}_o^{N_f}).$$

4.2.2.3 Stacked Attention Model.

As aforementioned, our coarse decoder generates words based on the global image features. However, in many cases, each word is only related to a small region of an image. Using the global image feature for word prediction could lead to sub-optimal results due to the noises introduced from the irrelevant regions for each prediction [5].

Therefore, the attention mechanism has been introduced to significantly improve the performance of image captioning. It typically produces a spatial map highlighting image regions relevant to each predicted word. In this research, to extract more fine-grained visual information for word prediction, we adopt a stacked attention model to filter out noises gradually and pinpoint the regions that are highly relevant to the word prediction. In each fine stage i , our attention model operates on both image features $\mathbf{V}$ and attention weights $\boldsymbol{\alpha}_t^{i-1}$ from the preceding stage.

Formally, for the time step t of stage i , the stacked attention model is defined as:

$$g(\mathbf{V}, \boldsymbol{\alpha}_t^{i-1}, h_t^{i-1}) = \sum_{n=0}^{k \times k - 1} \alpha_t^{i,n} \cdot (\mathbf{W}_{v\alpha}^i V_n + \mathbf{b}_{v\alpha}^i) \quad (4.5)$$

where $\alpha_t^{i,n}$ corresponds to the attention probability of each image region. We compute the attention probability $\alpha_t^{i,n}$ as follows:

$$\boldsymbol{\alpha}_t^i = \text{Softmax}(\mathbf{W}_\alpha^i \mathbf{A}_t^i + \mathbf{b}_\alpha^i) \quad (4.6)$$

$$\mathbf{A}_t^{i,n} = \tanh(\mathbf{W}_{va}^i V_n + \mathbf{W}_{ha}^i \bar{h}_t^{i-1}) \quad (4.7)$$

$$\bar{h}_t^{i-1} = h_t^{i-1} + \sum_{n=0}^{k \times k - 1} \alpha_t^{i-1,n} \cdot (\mathbf{W}_{v\alpha}^{i-1} V_n + \mathbf{b}_{v\alpha}^{i-1}) \quad (4.8)$$

where h_t^{i-1} is the updated hidden state of $\text{LSTM}_{\text{fine}}^{i-1}$, which is added to the aggregated image features to form a new hidden representation $\bar{h}_t^{i-1}$. Note that when $i = 1$, we set $\boldsymbol{\alpha}_t^0$ to zero.

4.2.3 Learning

The coarse-to-fine approach described above results in a deep architecture. Training such a deep network can be prone to the vanishing gradient problem, where the magnitude of gradients decreases in strength when back-propagated through

multiple intermediate layers. A natural approach to address this problem is to incorporate supervised training objectives into the intermediate layers. Each stage of the coarse-to-fine sentence decoder is trained to predict the words repeatedly. We first train the network by defining a loss function for each stage i that minimizes the cross-entropy (XE) loss, *i.e.*,

$$\mathcal{L}_{\text{XE}}^i(\theta_{0:i}) = - \sum_{t=0}^{T-1} \log(p_{\theta_{0:i}}(Y_t \mid Y_{0:t-1}, \mathbf{I})), \quad (4.9)$$

where the Y_t is the ground-truth word, and $\theta_{0:i}$ is the parameters up to the stage- i decoder. By adding the losses at each stage i , we obtain the overall learning objective for the full architecture:

$$\begin{aligned} \mathcal{L}_{\text{XE}}(\theta) &= \sum_{i=0}^{N_f} \mathcal{L}_{\text{XE}}^i(\theta_{0:i}) \\ &= - \sum_{i=0}^{N_f} \sum_{t=0}^{T-1} \log(p_{\theta_{0:i}}(Y_t \mid Y_{0:t-1}, \mathbf{I})) \end{aligned} \quad (4.10)$$

where $p_{\theta_{0:i}}(Y_t \mid Y_{0:t-1}, \mathbf{I})$ is the output probability of word Y_t given by the LSTM^{*i*} decoder. We share the weights of the models across all time steps.

However, training with the loss function of Eq. (4.10) is not sufficient. As mentioned in Section 1, the existing log-likelihood training methods have the problem of the discrepancy between their training and testing modes, where the model is often trained with scheduled sampling, while in testing, greed decoding or beam search is commonly used to get higher scores. Besides, the log-likelihood score of the prediction does not correlate well with the standard evaluation metrics such as BLEU and CIDEr. Many researchers have explored in the direction of optimizing the image captioning model with the evaluation metrics (*e.g.*, CIDEr in [28]). To optimise the evaluation metrics during each stage, we consider the image caption generation process as a reinforcement learning problem, *i.e.*, given an environment (previous states), we want to get an agent (*e.g.*, RNN, LSTM or GRU) to look

at the environment (image features, hidden states, and previous words), and make an action (the prediction of the next word). After generating a complete sentence, the agent will observe a sentence-level reward and update its internal state.

We cast our generative model in the reinforcement learning terminology as in [28, 64]. The LSTM-based decoder of each stage can be viewed as an agent that interacts with the external environment. The policy network parametrized by $\theta_{0:i}$ defines a policy $p_{\theta_{0:i}}$, which receives a state (preceding outputs, internal state of LSTM and image features) and produces an action $\tilde{Y}_t^i \sim p_{\theta_{0:i}}$ which is the prediction of the next word sampled from the LSTM at time step t . Once we have a complete predicted sentence $\tilde{\mathbf{Y}}^i$, the agent observes a reward $r(\tilde{\mathbf{Y}}^i)$ (*e.g.*, CIDEr score) of the sentence. The goal of RL-based training is to minimize the negative expected rewards (punishments) of multi-stages:

$$\mathcal{L}_{\text{RL}}(\theta) = - \sum_{i=1}^{N_f} \mathbb{E}_{\tilde{\mathbf{Y}}^i \sim p_{\theta_{0:i}}} [r(\tilde{\mathbf{Y}}^i)] \approx - \sum_{i=1}^{N_f} r(\tilde{\mathbf{Y}}^i) \quad (4.11)$$

where $\tilde{\mathbf{Y}}^i = \{\tilde{Y}_0^i, \dots, \tilde{Y}_{T-1}^i\}$, and $\tilde{Y}_t^i$ is sampled from the stage i at time step t . $r(\tilde{\mathbf{Y}}^i)$ is calculated by comparing the generated sentence to the corresponding reference sentences using the standard evaluation metric. Note that we do not consider $i = 0$ in Eq. (4.11) as the coarse decoder does not have a preceding stage. After that, we calculate the expected gradient using the Monte-Carlo sample $\tilde{\mathbf{Y}}^i$ from $p_{\theta_{0:i}}$ as:

$$\nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta) = \sum_{i=1}^{N_f} \nabla_{\theta_{0:i}} \mathcal{L}_{\text{RL}}(\theta_{0:i}) \quad (4.12)$$

$$\approx - \sum_{i=1}^{N_f} r(\tilde{\mathbf{Y}}^i) \cdot \nabla_{\theta_{0:i}} \log p_{\theta_{0:i}}(\tilde{\mathbf{Y}}^i) \quad (4.13)$$

To reduce the variance of the gradient estimate in Eq. (4.13), we follow the REINFORCE approach from SCST [28] to approximate Eq. (4.13) as:

$$\nabla_{\theta} \mathcal{L}_{\text{RL}}(\theta) \approx - \sum_{i=1}^{N_f} \Delta r(\tilde{\mathbf{Y}}^i) \cdot \nabla_{\theta_{0:i}} \log p_{\theta_{0:i}}(\tilde{\mathbf{Y}}^i) \quad (4.14)$$

where $\Delta r(\tilde{\mathbf{Y}}^i)$ is the relative reward which can reduce the variance of the gradient estimate. The principal idea of our RL-based coarse-to-fine learning approach is to baseline the REINFORCE algorithm with the reward $r(\hat{\mathbf{Y}}^i)$ obtained in each stage under the inference algorithm at test time, as well as the reward $r(\tilde{\mathbf{Y}}^{i-1})$ obtained by its preceding decoder at train time. Particularly, $\Delta r(\tilde{\mathbf{Y}}^i)$ is defined as:

$$\Delta r(\tilde{\mathbf{Y}}^i) = \left[r(\tilde{\mathbf{Y}}^i) - r(\hat{\mathbf{Y}}^i) \right] + \left[r(\tilde{\mathbf{Y}}^i) - r(\tilde{\mathbf{Y}}^{i-1}) \right] \quad (4.15)$$

where $\tilde{\mathbf{Y}}^i$ is a sampled caption of the i -th stage and $\hat{\mathbf{Y}}^i$ is obtained by the conventional greedy decoding. The first term in Eq. (4.15) tends to increase the probability of the samples of stage i that score higher than the results of stage i at test-mode (greedy decoding). In other words, we suppress those samples that have worse scores than the greedy decoding results. The second term increases the probability of the samples from stage i that outperform the samples from stage $i - 1$, and suppresses the inferior samples.

4.3 Experiments

In this section, we first describe the datasets used in our experiments, and then introduce the baseline methods for comparisons and the implementation details followed by the detailed results. We report all the results using MSCOCO caption evaluation tool¹.

¹<https://github.com/tylin/coco-caption>

4.3.1 Datasets and Setting

We evaluate the proposed approach on the MSCOCO dataset. The dataset contains 123,000 images, where each image has five reference captions. We follow the setting of [25] by using 5,000 images for offline validation and 5,000 images for offline testing. The widely used BLEU, METEOR, ROUGE, CIDEr, and SPICE scores are used to measure the quality of the generated captions. We further test on the MSCOCO test set consisting of 40,775 images, and then conduct the online comparison against the state-of-the-art via the online MSCOCO evaluation server.

4.3.2 Baseline Approaches for Comparisons

To gain insight into the effectiveness of our proposed approach, we compare the following models with each other:

- **LSTM and LSTM_{3 layers}.** We implement a one layer LSTM-based image captioning model based on the framework proposed by [112]. We also add two additional LSTM networks after the one layer LSTM model, which is named as LSTM_{3 layers}. We first train these two models with XE loss, and then optimize the CIDEr metric with SCST [28].
- **LSTM+ATT_{Soft} and LSTM+ATT_{Top-Down}.** We implement two types of visual attention-based image captioning models: the Soft-attention model (LSTM + ATT_{Soft}) proposed by [60] and the Top-Down attention model (LSTM+ATT_{Top-Down}) proposed by [119]. We encode the image with ResNet-101 and apply the spatially adaptive pooling to get a fixed-size output of $14 \times 14 \times 2048$. At each time step, the attention model produces an attention mask over the 196 spatial locations. LSTM+ATT_{Top-Down} consists of two LSTM networks, where the first LSTM takes the mean-pooled image feature

as input, and the second LSTM predicts the words based on the attended image features and the hidden state of the first LSTM. Similarly, we also train these two models with XE Loss and the RL-based sentence-level metric.

- **Stack-Cap and Stack-Cap***. Stack-Cap is our proposed method and Stack-Cap* is a simplified version. In particular, Stack-Cap* incorporates the multiple attention models into LSTM₃ layers. Here we treat the first LSTM as the coarse decoder, and the subsequent two attention-based LSTM networks ($N_f = 2$) as the fine decoders. Stack-Cap has the architecture similar to Stack-Cap*, except that it applies the proposed stacked attention model instead of the independent attention model. We train these two models (Stack-Cap* and Stack-Cap) with the proposed coarse-to-fine (C2F) learning approach.

4.3.3 Implementation Details

In this work, we set the number of hidden units of each LSTM to 512, the number of hidden units in the attention layer to 512, and the vocabulary size of the word embedding to 9,487. In our implementation, the parameters are randomly initialized except the image CNN, for which we encode the full image with the ResNet-101 pre-trained on ImageNet.

We first train our model under the cross-entropy cost using Adam [110] optimizer with an initial learning rate of 4×10^{-4} and a momentum parameter of 0.9. After that, we run the proposed RL-based approach on the just trained model to be optimized for the CIDEr metric. During this stage, we use Adam with a learning rate of 5×10^{-5} . After each epoch, we evaluate the model on the validation set and select the model with the best CIDEr score for testing. During testing, we apply beam search which can increase the performance of greedy decoding. Unlike greedy decoding which keeps only a single hypothesis during decoding, Beam search keeps

$K > 1$ ($K = 5$ in our experiments) hypotheses that have the highest scores at each time step, and returns the hypothesis with the highest log probability at the end.

4.3.4 Quantitative Analysis

In this experiment, we first optimize the models with the standard cross-entropy (XE) loss. We report the performance of our model and the baselines on the Karpathy test split in Table 4.1. Note that all results are reported without fine-tuning of the ResNet-101.

TABLE 4.1: Performance comparisons on MSCOCO, where B@n is short for BLEU-n, M is short for METEOR, and C is short for CIDEr. All values are reported as the percentage (**Bold** numbers are the best results).

Approach	B@1	B@2	B@3	B@4	M	C
LSTM (XE)	72.1	54.8	39.6	28.5	24.3	91.4
LSTM _{3 layers} (XE)	70.5	53.1	38.9	28.3	23.2	85.7
LSTM+Att _{Soft} (XE)	73.8	57.2	43.1	33.0	25.7	101.0
LSTM+Att _{Top-Down} (XE)	74.9	58.6	44.5	33.3	25.8	103.4
Stack-Cap* (XE)	75.6	59.6	45.6	34.6	26.3	108.0
Stack-Cap (XE)	76.2	60.4	46.4	35.2	26.5	109.1

It can be seen from Table 4.1 that our coarse-to-fine image captioning model (Stack-Cap) achieves the best performances in all metrics. The two coarse-to-fine models, Stack-Cap and Stack-Cap*, give similar performance. Note that although these two coarse-to-fine models have the same number of LSTM units as LSTM_{3 layers}, directly adding two additional LSTM layers in LSTM_{3 layers} without intermediate supervision decreases the performance of LSTM as the model experiences overfitting. Our coarse-to-fine approach can optimize the network gradually with the intermediate supervision and avoid overfitting. We also observe that Soft attention (LSTM+ATT_{Soft}) and Top-Down attention (LSTM+ATT_{Top-Down}) can significantly improve the performance of image captioning. Our best model (Stack-Cap) with stacked attention networks outperforms the Stack-Cap*, which

demonstrates that adjusting the attention on the relevant visual clues progressively can generate better image descriptions.

TABLE 4.2: Performance comparisons with the baselines on MSCOCO Karpathy test split. Our Stack-Cap (C2F) model achieves significant gains across all metrics.

Approach	B@1	B@2	B@3	B@4	M	C
LSTM (CIDEr)	76.7	58.3	42.8	30.8	25.5	100.2
LSTM _{3 layers} (CIDEr)	73.0	56.1	41.1	29.9	25.1	95.9
LSTM+Att _{Soft} (CIDEr)	77.3	59.3	44.1	32.1	25.9	104.8
LSTM+Att _{Top-Down} (CIDEr)	76.7	60.4	45.6	33.9	26.5	112.7
Stack-Cap* (C2F)	77.9	61.6	46.7	35.0	26.9	115.9
Stack-Cap (C2F)	78.6	62.5	47.9	36.1	27.4	120.4

After optimizing the models with XE loss, we optimize them for the CIDEr metric with the RL-based algorithms. The performances of the four models optimized for CIDEr with the SCST [28] and the performances of two models optimized with the proposed coarse-to-fine (C2F) learning are also reported in Table 4.2. We can see that our Stack-Cap model obtains significant gains across all metrics.

Table 4.3 compares the results of our Stack-Cap (C2F) model with those of the existing methods on MSCOCO Karpathy test split, where Stack-Cap achieves the best performance in all metrics.

Table 4.4 reports the performance of our proposed Stack-Cap model trained with the coarse-to-fine learning on the official MSCOCO evaluation server². We can see that our approach achieves very competitive performance, compared to the state-of-the-art. Note that the results of SCST:Att2in (Ens. 4) are achieved by the ensemble of four models, while our results are generated by the single model.

²<https://competitions.codalab.org/competitions/3221>

TABLE 4.3: Comparisons of the image captioning performance of the existing methods on MSCOCO Karpaty test split. Our Stack-Cap (C2F) model with the coarse-to-fine learning achieves significant gains across all metrics.

Approach	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
Google NIC [112]	—	—	—	27.7	—	23.7	85.5	—
Hard-Attention [60]	70.7	49.2	34.4	24.3	23.9	—	—	—
Soft-Attention [60]	71.8	50.4	35.7	25.0	23.0	—	—	—
VAE [114]	72.0	52.0	37.0	28.0	24.0	—	90.0	—
Google NICv2 [18]	—	—	—	32.1	25.7	—	99.8	—
Attributes-CNN+RNN [21]	74.0	56.0	42.0	31.0	26.0	—	94.0	—
CNN _c +RHN [10]	72.3	55.3	41.3	30.6	25.2	—	98.9	18.3
PG-SPIDER-TAG [67]	75.4	59.1	44.5	33.2	25.7	55.0	101.3	—
Adaptive [120]	74.2	58.0	43.9	33.2	26.6	—	108.5	—
SCST:Att2im [28]	—	—	—	33.3	26.3	55.3	111.4	—
SCST:Att2im (Ens. 4) [28]	—	—	—	34.8	26.9	56.3	115.2	—
Stack-Cap (C2F)	78.6	62.5	47.9	36.1	27.4	56.9	120.4	20.9

TABLE 4.4: Leaderboard of the published image captioning models (as of 10/09/2017) on the online MSCOCO test server. Our single Stack-Cap model trained with the coarse-to-fine learning yields comparable performance with the state-of-the-art approaches on all reported metrics.

Approach	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Google NIC	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	53.0	68.2	94.3	94.6
Hard-Attention	70.5	88.1	52.8	77.9	38.3	65.8	27.7	53.7	24.1	32.2	51.6	65.4	86.5	89.3
PG-SPIDER-TAG	75.1	91.6	59.1	84.2	44.5	73.8	33.1	62.4	25.5	33.9	55.1	69.4	104.2	107.1
Adaptive	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	55.0	70.5	104.2	105.9
SCST:Att2in (Ens. 4)	78.1	93.1	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
Ours: Stack-Cap (C2F)	77.8	93.2	61.6	86.1	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3

FIGURE 4.3: Visualizations of the generated captions and image attention maps on MSCOCO. Ground-Truth (GT) descriptions and the generated description of each stage are shown for each example. The columns from left to right correspond to the outputs of the three LSTM decoders from coarse to fine (coarse: *black*, refined: *purple*, final: *red*).

4.3.5 Qualitative Analysis

To demonstrate that using the proposed coarse-to-fine approach can generate better image descriptions stage-by-stage that correlate well with the adaptively attended regions, we visualize the spatial attention weight for word in the generated captions. We upsample the attention weights by a factor of 16 and apply a Gaussian filter to make it the same size as the input image, and stack all the upsampled spatial attention maps into the original input image.

Figure 4.3 shows some generated captions. By reasoning via multiple attention

layers progressively, the Stack-Cap model can gradually filter out noises and pinpoint the regions that are highly relevant to the current word prediction. We can find that our Stack-Cap model learns alignments that correspond strongly with human intuition. Taking the first image as an example, compared with the caption generated in the coarse stage, the first refined caption generated by the first fine decoder contains “*dog*,” and the second fine decoder not only produces “*dog*,” but also identifies “*umbrella*.”

Besides, our approach can generate more descriptive sentences. For example, the attention visualizations of the *jets* image show that the Stack-Cap model can query the relationship of those “*jets*” as well as the *long trail of smoke* behind them, as there are strong attention weights that encompass this salient region. This, together with other examples, suggests that the stacked attention can more effectively explore the visual information for sequence prediction. In other words, our approach via the stacked attention can consider visual information in the image from coarse to fine, aligning well with the human visual system, where we usually use a coarse-to-fine procedure to understand pictures.

4.4 Summary

In this chapter, we have presented a coarse-to-fine image captioning model which utilizes a stacked visual attention model in conjunction with multiple LSTM networks to achieve better image descriptions. Unlike the conventional one-stage models, our approach allows generating captions from coarse to fine, which we found to be very beneficial for image captioning. Our model achieves comparable performance with the state-of-the-art approach using ensemble on the online MSCOCO test server.

Chapter 5

Improving Textual-Visual Cross-Modal Retrieval with Generative Models

5.1 Introduction

As we are entering the era of big data, data from different modalities such as text, image, and video are growing at an unprecedented rate. Such multi-modal data exhibit heterogeneous properties, making it difficult for users to search information of interest effectively and efficiently. This chapter focuses on the problem in multi-modal information retrieval, which is to retrieve the images (resp. texts) that are relevant to a given textual (resp. image) query. The fundamental challenge in cross-modal retrieval lies in the heterogeneity of different modalities of data. Thus, the learning of a common representation shared by data with different modalities plays the key role in cross-modal retrieval.

In recent years, a great deal of research has been devoted to bridge the heterogeneity gap between different modalities [11, 25, 31, 83, 84, 121–123]. For textual-visual

FIGURE 5.1: Conceptual illustration of our proposed cross-modal feature embedding with generative models. The cross-modal retrievals (*Image-to-Text* and *Text-to-Image*) are shown in different color. The two blue boxes are cross-modal data, and the generated data are shown in two dashed yellow clouds.

cross-modal embedding, the common way is to first encode individual modalities into their respective features, and then map them into a common semantic space, which is often optimized via a ranking loss that encourages the similarity of the mapped features of ground-truth image-text pairs to be greater than that of any other negative pair. Once the common representation is obtained, the relevance / similarity between the two modalities can be easily measured by computing the distance (*e.g.* l_2) between their representations in the common space.

Although the feature representations in the learned common representation space have been successfully used to describe high-level semantic concepts of multi-modal data, they are not sufficient to retrieve images with detailed local similarity (*e.g.*, spatial layout) or sentences with word-level similarity. In contrast, as humans, we can relate a textual (resp. image) query to relevant images (resp. texts) more

accurately, if we pay more attention to the finer details of the images (resp. texts). In other words, if we can ground the representation of one modality to the objects in the other modality, we can learn a better mapping.

Inspired by this concept, in this work, we propose to incorporate generative models into textual-visual feature embedding for cross-modal retrieval. In particular, in addition to the conventional cross-modal feature embedding at the global semantic level, we also introduce an additional cross-modal feature embedding at the local level, which is *grounded* by two generative models: image-to-text and text-to-image. Figure 5.1 illustrates the concept of our proposed cross-modal feature embedding with generative models at high level, which includes three learning steps: *look*, *imagine*, and *match*. Given a query in image or text, we first *look* at the query to extract an *abstract* representation. Then, we *imagine* what the target item (text or image) in the other modality should look like, and get a more concrete *grounded* representation. We accomplish this by asking the representation of one modality (to be estimated) to generate the item in the other modality, and comparing the generated items with gold standards. After that, we *match* the right image-text pairs using the relevance score which is calculated based on a combination of *grounded* and *abstract* representations.

The contributions of this work are twofold. First, we incorporate two generative models into the conventional textual-visual feature embedding, which is able to learn concrete *grounded* representations that capture the detailed similarity between the two modalities. Second, we conduct extensive experimentations on the benchmark dataset, MSCOCO. Our empirical results demonstrate that the combination of the *grounded* and the *abstract* representations can significantly improve the state-of-the-art performance on cross-modal image-caption retrieval.

FIGURE 5.2: The proposed generative cross-modal learning framework (GXN). The entire framework consists of three training paths: cross-modal feature embedding (the entire upper part), image-to-text generative feature learning (the blue path), and text-to-image generative adversarial feature learning (the green path). It includes six networks: two sentence encoders $\text{RNN}_{\text{Enc}}^h$ (dark green) and $\text{RNN}_{\text{Enc}}^l$ (light green), one image encoder CNN_{Enc} (blue), one sentence decoder RNN_{Dec} , one image decoder CNN_{Dec} and one discriminator D_i .

5.2 Methodology

5.2.1 System Overview

Figure 5.2 shows the overall architecture for the proposed generative cross-modal feature learning framework, named GXN. The entire system consists of three training paths: multi-modal feature embedding (the entire upper part), image-to-text generative feature learning (the blue path), and text-to-image generative adversarial feature learning (the green path). The first path is similar to the existing cross-modal feature embedding that maps different modality features into a common space. However, the difference here is that we use two branches of feature embedding, *i.e.*, making the embedded visual feature v_h (resp. v_l) and the textual feature t_h (resp. t_l) closer. We consider (v_h, t_h) as high-level abstract features and (v_l, t_l) as detailed grounded features. The grounded features will be used and regularized in the other two generative feature learning paths. The entire first training path mainly includes one image encoder CNN_{Enc} and two sentence encoders $\text{RNN}_{\text{Enc}}^h$ and $\text{RNN}_{\text{Enc}}^l$.

The second training path (the blue path) is to generate a sentence from the embedded generative visual feature v_l . It consists of the image encoder CNN_{Enc} and a sentence detector RNN_{Dec} . With a proper loss against ground-truth sentences, the grounded feature v_l will be adjusted via back propagation. The third training path (the green path) is to generate an image from the textual feature t_l . Here we adopt the generative adversarial model, which comprises a generator / decoder CNN_{Dec} and a discriminator D_i .

Overall, through these two paths of cross-modal generative feature learning, we hope to learn powerful cross-modal feature representations. During the testing stage, $\{v_h, v_l\}$ and $\{t_h, t_l\}$ will be used as the final feature representations for cross-modal retrieval, although the proposed GXN also produces other byproducts such as image-to-text generation and text-to-image generation, which are not the main focus of this work. In the following, we describe each of the three training paths in detail.

5.2.2 Cross-modal Feature Embedding

We follow the common cross-modal feature embedding approach to embed the representations of the image and the caption into a common space, and then use a pairwise ranking loss to learn the model parameters [84]. In particular, given an image-caption pair (i, c) , where i is the image and $c = (w_0, \dots, w_{T-1})$ is the corresponding description with w_i being the one-hot word encoding, we encode a caption by embedding each word in c into a distributed representation using $\mathbf{W}_e w_i$, where $\mathbf{W}_e$ is a shared word embedding matrix to be learned. $\mathbf{W}_e$ can be initialized randomly or using pre-trained embeddings like word2vec [124]. Then we use two sequential sentence encoders (*e.g.*, GRU) to get the sentence representations. As for image encoding, we use a CNN that is pre-trained on ImageNet. More formally, we formulate the embedding and mapping of each modality as:

$$\begin{aligned} v_k &= P_v^k(\text{CNN}_{\text{Enc}}(i; \theta_i)) \\ t_k &= P_t^k(\text{RNN}_{\text{Enc}}^k(c; \theta_c^k)) \end{aligned}, \quad k \in \{h, l\} \quad (5.1)$$

where θ_i and θ_c^k are the parameters of the image and caption encoders, P_v^k and P_t^k are the linear transformation functions which map the encoded vectors into a common embedding space, and v_k and t_k are the resulting mapped vectors for the image and the caption, respectively.

We first consider the same pairwise ranking loss proposed in [25, 83, 84, 125]. We refer (i, c) as positive pairs and denote the negative samples by i' and c' , where i' goes over images not described by c and c' goes over captions that do not describe i . We want the objective function to encourage the similarity of ground truth caption-image pairs to be greater than that of all other negative pairs. We, therefore, optimize the ranking loss of $\mathcal{L}_{\text{Rank}} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}_{\text{R}}(i_n, c_n)$, where the single sample ranking loss $\mathcal{L}_{\text{R}}$ is defined as:

$$\begin{aligned} \mathcal{L}_{\text{R}} &= \sum_{t'} [\alpha - s(t, v) + s(t', v)]_+ + \\ &\quad \sum_{v'} [\alpha - s(t, v) + s(t, v')]_+ \end{aligned} \quad (5.2)$$

where α is a margin, $s(t, v) = -\|\max(0, v - t)\|^2$ is the order-violation penalty [84] used as a similarity, t' and v' denote the representations of the negative samples. Here, $[x]_+$ represents $\max(x, 0)$.

Considering we have two branches of cross-modal feature embedding, which result in two pairs of cross-modal features: the abstract features (t_h, v_h) and the grounded features (t_l, v_l) , we modify the ranking loss as:

$$\begin{aligned} \mathcal{L}_{\text{R}^+} &= \sum_{t'} [\alpha - s^*(t_{h,l}, v_{h,l}) + s^*(t'_{h,l}, v_{h,l})]_+ + \\ &\quad \sum_{v'} [\alpha - s^*(t_{h,l}, v_{h,l}) + s^*(t_{h,l}, v'_{h,l})]_+ \end{aligned} \quad (5.3)$$

where $s^*(t_{h,l}, v_{h,j}) = \lambda s(t_h, v_h) + (1 - \lambda)s(t_l, v_l)$ is a combined score with λ being the tradeoff weight.

5.2.3 Image-to-text Generative Feature Learning

For the image-to-text training path (i2t, blue path in Figure 5.2), our goal is to encourage the *grounded* visual feature v_l to be able to generate sentences that are similar to the ground-truth captions. In particular, we first encode the image with CNN_{Enc} , and then decode the grounded visual feature into a sentence with RNN_{Dec} . Like the traditional RNN-based text generation models, we first train our model on a cross-entropy (XE) loss defined as:

$$\mathcal{L}_{\text{xe}} = - \sum_{t=0}^{T-1} \log p_{\theta_t}(w_t | w_{0:t-1}, v_l; \theta_t) \quad (5.4)$$

where w_t is the ground-truth word, $p_{\theta_t}(w_t | w_{0:t-1}, v_l)$ is the output probability of word w_t given by the decoder with parameter θ_t .

However, the XE loss is a word-level cost, and models trained on this suffer from the exposure bias problem [59] and the loss-evaluation mismatch problem [11, 64]. Thus we further employ a loss that takes the entire sequence into account. Specifically, to directly optimize the sentence-level metrics, we optimize our model by minimizing the negative expected reward given by:

$$\mathcal{L}_{\text{rl}} = -\mathbb{E}_{\tilde{c} \sim p_{\theta_t}}[r(\tilde{c})] \quad (5.5)$$

where $\tilde{c} = (\tilde{w}_0, \dots, \tilde{w}_{T-1})$ is the word sequence sampled from the decoder, $r(\tilde{c})$ is the reward calculated by comparing the generated sentence with the corresponding reference sentences using a standard evaluation metric like BLEU [65] or CIDEr [66]. Following the reinforcement learning (RL) approach described in [11, 28], the expected gradients of Eq. (5.5) using Monte-Carlo sample $\tilde{c}$ from

p_{θ_t} can be approximated as:

$$\begin{aligned}\nabla_{\theta_t} \mathcal{L}_{\text{rl}} &= -\mathbb{E}_{\tilde{c} \sim p_{\theta_t}} [r(\tilde{c}) \cdot \nabla_{\theta_t} \log p_{\theta_t}(\tilde{c})] \\ &\approx -r(\tilde{c}) \nabla_{\theta_t} \log p_{\theta_t}(\tilde{c}) \\ &\approx -(r(\tilde{c}) - r_b) \nabla_{\theta_t} \log p_{\theta_t}(\tilde{c})\end{aligned}\tag{5.6}$$

where r_b is the baseline estimator used to reduce the variance without changing the expected gradient. In our model, we use the inference process reward as the baseline.

During the early stage of training, optimization of Eq. (5.6) alone does not ensure the readability and fluency of the generated caption [126]. To deal with this, we use a mixture of XE and RL losses:

$$\mathcal{L}_{\text{xe+rl}} = (1 - \gamma) \mathcal{L}_{\text{xe}} + \gamma \mathcal{L}_{\text{rl}}\tag{5.7}$$

where γ is a tuning parameter used to balance the two losses. Eq. (5.7) improves results on the metric used to compute the reward through the reinforcement loss but also ensures better readability and fluency due to the XE loss. For annealing and faster convergence, we start with optimizing XE loss in Eq. (5.4), and then move to optimizing the joint loss in Eq. (5.7).

5.2.4 Text-to-image Generative Adversarial Feature Learning

For the text-to-image training path (t2i, the green path in Figure 5.2), our goal is to encourage the grounded text feature t_l to be able to generate an image that is similar to the ground-truth one. However, unlike the image-to-text path in Section 5.2.3, where the model is trained to predict the word conditioned on image

and history words, the reverse path suffers from the highly multi-modal distribution of images conditioned on a text representation.

The natural way to model such a conditional distribution is to use a conditional GAN [73, 74], which consists of a discriminator and a generator. The discriminator is trained to distinguish the real samples $\langle \text{real image, true caption} \rangle$ from the generated samples of $\langle \text{fake image, true caption} \rangle$ as well as samples of $\langle \text{real image, wrong caption} \rangle$. Specifically, the discriminator D_i and the generator G_i (CNN_{Dec} in Figure 5.2) play the min-max game on the following value function $V(D_i, G_i)$:

$$\min_{G_i} \max_{D_i} V(D_i, G_i) = \mathcal{L}_{D_i} + \mathcal{L}_{G_i}. \quad (5.8)$$

The discriminator loss $\mathcal{L}_{D_i}$ and the generator loss $\mathcal{L}_{G_i}$ are defined as:

$$\begin{aligned} \mathcal{L}_{D_i} = & \mathbb{E}_{i \sim p_{\text{data}}} [\log D_i(i, t_l)] + \beta_f \mathbb{E}_{\hat{i} \sim p_G} [\log(1 - D_i(\hat{i}, t_l))] + \\ & \beta_w \mathbb{E}_{i \sim p_{\text{data}}} [\log(1 - D_i(i, t'_l))] \end{aligned} \quad (5.9)$$

$$\mathcal{L}_{G_i} = \mathbb{E}_{\hat{i} \sim p_G} [\log(1 - D_i(\hat{i}, t_l))] \quad (5.10)$$

where t_l and t'_l denote the encoded grounded feature vectors for a matched and a mismatched captions, respectively, i is the matched real image from the true data distribution p_{data} , β_f and β_w are the tuning parameters, and $\hat{i} = G_i(z, t_l)$ is the generated image by the generator G_i conditioned on t_l and a noise sample z . The variable z is sampled from a fixed distribution (*e.g.*, uniform or Gaussian distribution). In implementation, we compress t_l to a lower dimension and then combine it with z .

However, directly combining t_l with z cannot produce satisfactory results. This is because of the limited amount of data and the unsmoothness between t_l and z . Thus, we introduce another variable t_c , which is sampled from a Gaussian distribution of $\mathcal{N}(\mu(\varphi(t_l)), \sigma(\varphi(t_l)))$ [127], where $\mu(\varphi(t_l))$ and $\sigma(\varphi(t_l))$ are the mean and the standard deviation of t_l , $\varphi(t_l)$ compresses t_l to a lower dimension. We now

generate the image conditioned on z and t_c with $\hat{i} = G_i(z, t_c)$. The discriminator loss $\mathcal{L}_{D_i}$ and the generator loss $\mathcal{L}_{G_i}$ are then modified to:

$$\begin{aligned} \mathcal{L}_{D_i} = & \mathbb{E}_{i \sim p_{\text{data}}} [\log D_i(i, t_l)] + \beta_f \mathbb{E}_{\hat{i} \sim p_G} [\log(1 - D_i(\hat{i}, t_l))] + \\ & \beta_w \mathbb{E}_{i \sim p_{\text{data}}} [\log(1 - D_i(i, t'_l))] \end{aligned} \quad (5.11)$$

$$\begin{aligned} \mathcal{L}_{G_i} = & \mathbb{E}_{\hat{i} \sim p_G} [\log(1 - D_i(\hat{i}, t_l))] + \\ & \beta_s \mathcal{D}_{\text{KL}}(\mathcal{N}(\mu(\varphi(t_l)), \sigma(\varphi(t_l))) \parallel \mathcal{N}(0, 1)) \end{aligned} \quad (5.12)$$

where β_f , β_w and β_s are the tuning parameters, and the KL-divergence term is to enforce the smoothness of the latent data manifold.

Alg. 1 summarizes the entire training procedure.

5.3 Experiments

5.3.1 Dataset and Implementation Details

We evaluate our approach on the MSCOCO dataset [15]. For cross-modal retrieval, we use the setting of [25], which contains 113,287 training images with five captions each, 5,000 images for validation, and 5,000 images for testing. We experiment with two image encoders: VGG19 [98] and ResNet152 [41]. For VGG19, we extract the features from the penultimate fully connected layer. For ResNet152, we obtain the global image feature by taking a mean-pooling over the last spatial image features. The dimensions of the image feature vectors is 4096 for VGG19 and 2048 for ResNet152. As for text preprocessing, we convert all sentences to lower case, resulting in a vocabulary of 27,012 words.

We set the word embedding size to 300 and the dimensionality of the joint embedding space to 1024. For the sentence encoders, we use a bi-directional GRU-based encoder to get the *abstract* feature representation t_h and one GRU-based encoder

Algorithm 1 GXN training procedure.

Input: Positive image i , negative image i' , positive text c , negative text c' , number of training batch steps S

- 1: **for** $n = 1 : S$ **do**
- 2: /*Look*/
- 3: Draw image-caption pairs: (i, c) , i' and c' .
- 4: $v_h, v_l, v'_h, v'_l \leftarrow i, i'$ {Image encoding}
- 5: $t_h, t_l, t'_h, t'_l \leftarrow c, c'$ {Text encoding}
- 6: Update parameters with Gen_{i2t}-GXN
- 7: Update parameters with Gen_{t2i}-GXN
- 8: **end for**

Function: Gen_{i2t}-GXN

- 1: /*Imagine*/
- 2: $\hat{c} = \text{RNN}_{\text{Dec}}(v_l, c)$ {Scheduled sampling}
- 3: Compute XE loss $\mathcal{L}_{\text{xe}}$ using (5.4).
- 4: $\tilde{c} \leftarrow \text{RNN}_{\text{Dec}}(v_l)$ {Sampling}
- 5: $\bar{c} \leftarrow \text{RNN}_{\text{Dec}}(v_l)$ {Greedy decoding}
- 6: Compute RL loss $\mathcal{L}_{\text{rl}}$ using (5.5).
- 7: Update model parameters by descending stochastic gradient of (5.7) with $r_b = r(\bar{c})$ (see (5.6)).
- 8: /*Match*/
- 9: Update model parameters using (5.3).

Function: Gen_{t2i}-GXN

- 1: /*Imagine*/
- 2: $t_c \sim \mathcal{N}(\mu(\varphi(t_l)), \sigma(\varphi(t_l)))$
- 3: $\hat{i} = G_i(z, t_c)$
- 4: Update image discriminator D_i using (5.11).
- 5: Update image generator G_i using (5.12).
- 6: /*Match*/
- 7: Update model parameters using (5.3).

to get the *grounded* feature representation t_l . The number of hidden units of both GRUs is set to 1024. For the sentence decoder, we adopt a one-layer GRU-based decoder which has the same hidden dimensions as the two GRU-based encoders. During the RL training, we use CIDEr score as the sentence-level reward. We set $\beta_f = 0.5$, $\beta_w = 0.5$ and $\beta_s = 2.0$ in Eq. (5.11) and (5.12), margin α and λ in Eq. (5.3) to be 0.05 and 0.5 respectively, and γ in Eq. (5.7) is increased gradually based on the epoch from 0.05 to 0.95. The output size of the image decoder CNN_{Dec} is $64 \times 64 \times 3$, and the real image is resized before inputting to the discriminator. All the modules are randomly initialized before training except for the

CNN encoder and decoder. Dropout and batch normalization are used in all our experiments. We use Adam [110] for optimization with a mini-batch size of 128 in all our experiments. The initial learning rate is 0.0002, and the momentum is 0.9.

For evaluation, we use the same measures as those in [84], *i.e.*, R@K, defined as the percentage of queries in which the ground-truth matchings are contained in the first K retrieved results. The higher value of R@K means better performance. Another metric we use is Med r , which is the median rank of the first retrieved ground-truth sentence or image. The lower its value, the better. We also compute another score, denoted as ‘**Sum**’, to evaluate the overall performance for cross-modal retrieval, which is the summation of all R@1 and R@10 scores defined as follows:

$$\text{Sum} = \underbrace{\text{R@1} + \text{R@10}}_{\text{Image-to-Text}} + \underbrace{\text{R@1} + \text{R@10}}_{\text{Text-to-Image}} \quad (5.13)$$

In addition, we evaluate the quality of the generated captions with the standard evaluation metrics: CIDEr and BLEU- n . BLEU- n rates the quality of the retrieved captions by comparing n -grams of the candidate with the n -grams of the five gold captions and count the number of matches. CIDEr is a consensus-based metric which is more correlated with human assessment of caption quality.

5.3.2 Baseline Approaches for Comparisons

GRU (VGG19) and GRU_{Bi} (VGG19): These two baselines use the pre-trained VGG19 as the image encoder. GRU (VGG19) adopts a one layer GRU as the sentence encoder, while GRU_{Bi} (VGG19) adopts a bi-directional GRU as the sentence encoder. These two models are trained using Eq. (5.2).

GXN (ResNet152) and GXN (fine-tune): These two baselines use the same two GRU sentence encoders as our proposed GXN framework, but without the generation components. In other words, they only contain the cross-modal feature embedding training path using Eq. (5.3). Here, the pre-trained ResNet152 is

adopted as the image encoder. GXN (ResNet152) and GXN (fine-tune) refer to the models without or with fine-tuning ResNet152, respectively. The fine-tuned ResNet152 model is used as the image encoder for all other GXN models.

GXN (i2t, xe) and GXN (i2t, mix): These two GXN baseline models contain not only the cross-modal feature embedding training path but also the image-to-text generative training path. GXN (i2t, xe) and GXN (i2t, mix) are the two models optimized with Eq. (5.4) and (5.7), respectively.

GXN (t2i): This baseline model contains both the cross-modal feature embedding training path and the text-to-image generative training path, and is trained with Gen_{t2i} -GXN in Algorithm 1.

GXN (i2t+t2i): This is our proposed full GXN model containing all the three training paths. It is initialized with the trained parameters from GXN (i2t, mix) and GXN (t2i) and fine-tuned with Algorithm 1.

5.3.3 Quantitative Results

In this section, we present our quantitative results and analysis. To verify the effectiveness of our approach and to analyze the contribution of each component, we compare different baselines in Table 5.1 and 5.2. The comparison of our approach with the state-of-the-art methods is shown in Table 5.3.

Effect of a Better Text Encoder. The first two rows in Table 5.1 compare the effectiveness of the two sentence encoders. Compared with GRU (VGG19), GRU_{Bi} (VGG19) can make full use of the context information from both directions and achieve better performance, *i.e.*, GRU_{Bi} (VGG19) increases the caption retrieval R@1 from 51.4 to 53.6 and image retrieval R@1 from 39.1 to 40.0.

Effect of a Better Image Encoder. We further investigate the effect of image encoding model on the cross-modal feature embedding. By replacing the VGG19

TABLE 5.1: Cross-modal retrieval results on MSCOCO 1K-image test set (bold numbers are the best results).

Model	Image-to-Text			Text-to-Image		
	R@1	R@10	Med	R@1	R@10	Med
GRU(VGG19)	51.4	91.4	1.0	39.1	86.7	2.0
GRU _{Bi} (VGG19)	53.6	90.2	1.0	40.0	87.8	2.0
GXN(ResNet152)	59.4	94.7	1.0	47.0	92.6	2.0
GXN(fine-tune)	64.0	97.1	1.0	53.6	94.4	1.0
GXN(i2t,xe)	68.2	98.0	1.0	54.5	94.8	1.0
GXN(i2t,mix)	68.4	98.1	1.0	55.6	94.6	1.0
GXN(t2i)	67.1	98.3	1.0	56.5	94.8	1.0
GXN (i2t+t2i)	68.5	97.9	1.0	56.6	94.5	1.0

model in GRU_{Bi} (VGG19) with ResNet152, we achieve huge performance gains. The caption retrieval R@1 increases from 53.6 to 64.0, and the image retrieval R@1 increases from 40.0 to 53.6.

Effect of the Generative Models. We first consider the incorporation of the image-to-caption generation process into our GXN model. From Table 5.1, it can be seen that, compared with GXN (fine-tune), GXN (i2t, xe) achieves significantly better performance on the image-to-text retrieval. This validates our assumption that by combining the *abstract* representation with the *grounded* representation learned by caption generation (imagining), we can retrieve more relevant captions. Then, as we further enrich the model with the mixed RL+XE loss of Eq. (5.7), we observe further improvements (see GXN (i2t, mix)).

We also evaluate the effect of incorporating the text-to-image generation process into our GXN model. It can be seen from Table 5.1 that, compared with GXN (fine-tune), GXN (t2i) significantly improves the text-to-image retrieval performance. This is because the *grounded* text feature t_i is well learned via the text-to-image generation process (imagining). Although the image-to-text retrieval performance of GXN (t2i) is not as good as GXN (i2t, mix), it is still much better than GXN (fine-tune), which does not incorporate any generative process.

The final row in Table 5.1 shows the performance of our complete model, *i.e.*, GXN (i2t+t2i), which incorporates both image and text generations. We can see that GXN (i2t+t2i) achieves the best performances in general, having the advantages of both GXN (i2t, mix) and GXN (t2i).

Quality of the retrieved captions. For the image-to-text retrieval task in Table 5.1, Table 5.2 reports the quality of the retrieved captions using the sentence-level metrics, BLEU and CIDEr. Both BLEU and CIDEr have been shown to correlate well with human judgments [66]. As shown in Table 5.2, incorporating the generative models into GXN yields better results than GXN (fine-tune) that does not incorporate any generation process. Note that those scores are calculated over five reference sentences. This demonstrates that our proposed GXN model can retrieve captions that are closer to the ground-truth ones.

5.3.3.1 Comparisons with the State-of-the-art

Table 5.3 shows the comparisons of our cross-modal retrieval results on MSCOCO dataset with state-of-the-art methods. We can see that our framework achieves the best performance in all metrics, which clearly demonstrates the advantages of our model. To make our approach more convincing and generic, we also conduct experiments on Flickr30K dataset with results shown in Table 5.4.

5.3.4 Qualitative Results

In this section, we present a qualitative analysis of our GXN (i2t+t2i) framework on cross-modal retrieval.

Results of image-to-text retrieval. Figure 5.3 depicts some examples for image-to-text retrieval, where the results of VSE0 and VSE++ are adopted from [89]. We

Retrieved Captions:

- Ours:**
1. a young man doing a skateboard trick while others watch
 2. man doing a skate trick during a competition event with a audience
 3. skateboarders are doing tricks as a crowd watches
 4. a group of people doing skateboarding tricks on a car
 5. a boy riding on his skateboard at a skate park while other guys watch

VSE0 : Young skateboarder displaying skills on sidewalk near field

VSE++: Two young men are outside skateboarding together

Generated caption: A man doing a trick on a skateboard on a skate park

- GT**
- Some skateboarders doing tricks and people watching them
 - Boys skateboarding over a rusty car onto platforms, with an audience watching
 - A group of kids doing stunts at a skating event
 - Skateboarders are doing tricks as a crowd watches
 - A group of teens performing stunts at a skateboard park

FIGURE 5.3: Visual results of image-to-text retrieval, where the top-5 retrieved captions and the generated caption are shown in red color.

FIGURE 5.4: Visual results of text-to-image retrieval. 2nd row: retrieved images. 3rd row: image samples generated by our conditional GAN.

TABLE 5.2: Evaluating the quality of the retrieved captions on the MSCOCO 1K test set using the sentence-level metrics, where B@n is a short form for BLEU- n , and C is a short form for CIDEr. All values are reported in percentage. The 2nd column is the rank order of the retrieved caption.

Model	No.	B@1	B@2	B@3	B@4	C
GXN(fine-tune)	1	54.6	34.5	21.0	12.9	56.3
GXN(i2t,xe)	1	56.5	36.2	22.6	14.1	59.2
GXN(i2t,mix)	1	57.0	36.7	23.0	14.4	60.0
GXN(t2i)	1	56.0	36.0	22.4	14.3	58.8
GXN(t2i+t2i)	1	57.1	36.9	23.3	14.9	61.1
	2	55.8	35.8	22.4	13.7	58.3
	3	54.2	33.6	20.5	12.7	54.0
	4	53.1	32.9	19.9	11.9	51.2
	5	53.2	32.8	19.6	11.3	51.1

FIGURE 5.5: More visual results of cross-modal retrieval.

show the top-5 retrieved captions as well as the ground-truth captions. We can see that the retrieved captions of our model can better describe the query images.

Results of text-to-image retrieval. Figure 5.4 depicts some examples for text-to-image retrieval, where we show the top-5 retrieved images as well as the generated images. Compared to the ground-truth image and the retrieved images, although the generated images are of limited quality for complex multi-object scenes, they still contain certain plausible shapes, colors, and backgrounds. This suggests that our model can capture the complex underlying language-image relations.

Some more samples are shown in Figure 5.5. We show the retrieved and generated results for both image-to-text and text-to-image on the same image-caption pairs.

TABLE 5.3: Comparisons of the cross-modal retrieval results on MSCOCO dataset with the state-of-the-art methods. We mark the unpublished work with * symbol. Note that ‘Sum’ is the summation of the two R@1 scores and the two R@10 scores.

Model	Image-to-Text Retrieval			Text-to-Image Retrieval			Sum
	R@1	R@10	Med r	R@1	R@10	Med r	
	1K Test Images						
m-CNN [128]	42.8	84.1	2.0	32.6	82.8	3.0	242.3
HM-LSTM [92]	43.9	87.8	2.0	36.1	86.7	3.0	254.5
Order-embeddings [84]	46.7	88.9	2.0	38.9	85.9	2.0	260.4
DSPE+Fisher Vector [129]	50.1	89.2	-	39.6	86.9	-	265.8
sm-LSTM [93]	53.2	91.5	1.0	40.7	87.4	2.0	272.8
*VSE++ (ResNet152, fine-tune) [89]	64.7	95.9	1.0	52.0	92.0	1.0	304.6
GXN (i2t+t2i)	68.5	97.9	1.0	56.6	94.5	1.0	317.5
5K Test Images							
Order-embeddings [84]	23.3	65.0	5.0	18.0	57.6	7.0	163.9
*VSE++ (ResNet152, fine-tune) [89]	41.3	81.2	2.0	30.3	72.4	4.0	225.2
GXN(t2i+t2i)	42.0	84.7	2.0	31.7	74.6	3.0	233.0

TABLE 5.4: Experimental results on Flickr30K 1k image test set.

Model	Image-to-Text				Text-to-Image			
	R@1	R@10	Med	r	R@1	R@10	Med	r
*VSE++ [89]	52.9	87.2	1.0		39.6	79.5	2.0	
GXN (i2t+t2i)	56.8	89.6	1.0		41.5	80.1	2.0

FIGURE 5.6: Visualization of word embedding.

Results of word embedding. As a byproduct, a word embedding matrix $\mathbf{W}_e$ (mentioned at the beginning of Section 5.2.2) is also learned in our GXN models. We visualize the learned word embedding by projecting some selected word vectors into a 2-D space in Figure 5.6. We can see that compared with the embeddings learned from GXN (fine-tune), our GXN (i2t+t2i) can learn word embedding with more related visual meaning. For example, we find that words like ‘eats’ and ‘stares’ of GXN (i2t+t2i) are closer to each other compared to those of GXN (fine-tune). This is also consistent with the fact that when we ‘eat’ some food; we also tend to ‘stare’ at it.

5.4 Summary

In this chapter, we have proposed a novel cross-modal feature embedding framework for cross image-text retrieval. The uniqueness of our framework is that we incorporate the image-to-text and the text-to-image generative models into the conventional cross-modal feature embedding. We learn both the high-level abstract representation and the local grounded representation of multi-modal data in a max-margin learning-to-rank framework. Our framework significantly outperforms state-of-the-art methods for textural-visual cross-modal retrieval on MSCOCO dataset.

Chapter 6

Unpaired Image Captioning by Language Pivoting

This chapter, together with Chapter 7, aims to tackle the problem of unpaired image captioning. Two aspects are studied: pivot language-based solution, and scene graph-based solution. This chapter focuses on the first aspect.

6.1 Introduction

Recent several years have witnessed unprecedented advancements in automatic image caption generation. This progress can be attributed *(i)* to the invention of novel deep learning framework that learns to generate natural language descriptions of images in an end-to-end fashion, and *(ii)* to the availability of large annotated corpora of images paired with captions such as MSCOCO [15] to train these models. The dominant methods are based on an encoder-decoder framework, which uses a deep CNN to encode the image into a feature vector, and then use a RNN

to generate the caption from the encoded vector [25, 95, 112]. More recently, approaches of using attention mechanisms and reinforcement learning have dominated the MSCOCO captioning leaderboard [7, 11, 28].

Despite the impressive results achieved by the deep learning framework, one performance bottleneck is the availability of large paired datasets because neural image captioning models are generally *annotation-hungry* requiring a large amount of annotated image-caption pairs to achieve effective results [31]. However, in many applications, such large-scale annotations are not readily available, and are expensive and tedious to acquire. In these scenarios, unsupervised methods that can generate captions from unpaired data or semi-supervised methods that can exploit paired annotations from other domains or languages are highly desirable [130]. In this work, we pursue the later research avenue, where we assume that we have access to image-caption paired instances in one language (Chinese), and our goal is to transfer this knowledge to a target language (English) for which we do not have such image-caption paired datasets. We also assume that we have access to a separate source-target (Chinese-English) parallel corpus to help us with the transformation. In other words, we wish to use the source language (Chinese) as a pivot language to bridge the gap between an input image and a caption in the target language (English).

The concept of using a pivot language as an intermediary language has been studied previously in machine translation (MT) to translate between a resource-rich language and a resource-scarce language [131–134]. The translation task in this strategy is performed in two steps. A source-to-pivot MT system first translates a source sentence into the pivot language, which is, in turn, translated to the target language using a pivot-to-target MT system. Although related, image captioning with the help of a pivot language is fundamentally different from MT, since it involves putting together two different tasks – captioning and translation. In addition, the pivot-based *pipelined* approach to MT suffers from two major problems

when it comes to image captioning. First, the conventional pivot-based MT methods assume that the datasets for source-to-pivot and pivot-to-target translations come from the same (or similar) domain(s) with similar styles and word distributions. However, as it comes to image captioning, captions in the pivot language (Chinese) and sentences in the (Chinese-English) parallel corpus are quite different in styles and word distributions. For instance, MSCOCO captioning dataset mostly consists of images of a large scene with object instances (nouns), whereas parallel language corpora are more generic. Second, the errors made in the source-to-pivot translation get propagated to the pivot-to-target translation module in the pipelined approach.

In this work, we present an approach that can effectively capture the characteristics of an image captioner from the source language and align it to the target language using another source-target parallel corpus. More specifically, our pivot-based image captioning framework comprises an image captioner *image-to-pivot*, an encoder-decoder model that learns to describe images in the pivot language, and a *pivot-to-target* translation model, another encoder-decoder model that translates the sentence in pivot language to the target language, and these two models are trained on two separate datasets. We tackle the variations in writing styles and word distributions in the two datasets by adapting the language translation model to the captioning task. This is achieved by adapting both the encoder and the decoder of the pivot-to-target translation model. In particular, we regularize the word embeddings of the encoder (of pivot language) and the decoder (of target language) models to make them similar to image captions. We also introduce a joint training algorithm to connect the two models and enable them to interact with each other during training. We use AIC-ICC [17] and AIC-MT [17] as the training datasets and two datasets (MSCOCO and Flickr30K [16]) as the validation datasets. The results show that our approach yields substantial gains over the baseline methods on the validation datasets.

6.2 Methodology

Let $D_{i,x} = \{(i, x)^{(n_i)}\}_{n_i=0}^{N_i-1}$ denote the dataset with N_i image-caption pairs, and $D_{x,y} = \{(x, y)^{(n_x)}\}_{n_x=0}^{N_x-1}$ denote the translation dataset with N_x source-target sentence pairs. For notational simplicity, we use i to denote an image instance as well as the image modality. Similarly, we use x to represent a source sentence as well as a source/pivot language (Chinese), and y to represent a target sentence and a target language (English). Our ultimate goal is to learn a mapping function to describe an image i with a caption y . Formally,

$$y \sim \arg \max_y \{P(y|i; \theta_{i \rightarrow y})\} \quad (6.1)$$

where $\theta_{i \rightarrow y}$ are the model parameters to be learned in the absence of any paired data, $i^{(n_i)} \leftrightarrow y^{(n_y)}$. We use the pivot language x to learn the mapping: $i \xrightarrow{\theta_{i \rightarrow x}} x \xrightarrow{\theta_{x \rightarrow y}} y$. Note that image-to-pivot ($D_{i,x}$) and pivot-to-target ($D_{x,y}$) in our setting are two distinct datasets with possibly no common elements.

FIGURE 6.1: Pictorial depiction of our pivot-based unpaired image captioning setting. Here, i , x , y , and $\hat{y}$ denote source image, pivot language sentence, target language sentence, and ground truth captions in the target language, respectively. We use a dashed line to denote that there is no parallel corpus available for the pair. Solid lines with arrows represent decoding directions. Dashed lines inside a language (circle) denote stylistic and distributional differences between caption and translation data.

Figure 6.1 illustrates our pivot-based image captioning approach. We have an image captioning model $P(x|i; \theta_{i \rightarrow x})$ to generate a caption in the pivot language from an image and a NMT model $P(y|x; \theta_{x \rightarrow y})$ to translate this caption into the target

language. In addition, we have an autoencoder in the target language $P(\hat{y}|\hat{y}; \theta_{\hat{y} \rightarrow \hat{y}})$ that guides the target language decoder to produce caption-like sentences. We train these components jointly so that they interact with each other. During inference, given an unseen image i to be described, we use the joint decoder:

$$y \sim \arg \max_y \{P(y|i; \theta_{i \rightarrow x}, \theta_{x \rightarrow y})\} \quad (6.2)$$

In the following, we first give an overview of neural methods for image captioning and machine translation using paired (parallel) data. Then, we present our approach that extends these standard models for unpaired image captioning with a pivot language.

Standard Image Captioning. For image captioning in the paired setting, the goal is to generate a caption $\tilde{x}$ from an image i such that $\tilde{x}$ is as similar to the ground truth caption x . We use $P_x(x|i; \theta_{i \rightarrow x})$ to denote a standard encoder-decoder based image captioning model with $\theta_{i \rightarrow x}$ being the parameters. We first encode the given image to the image features v with a CNN-based image encoder: $v = \text{CNN}(i)$. Then, we predict the image description x from the global image feature v . The training objective is to maximize the probability of the ground truth caption words given the image:

$$\tilde{\theta}_{i \rightarrow x} = \arg \max_{\theta_{i \rightarrow x}} \{ \mathcal{L}_{i \rightarrow x} \} \quad (6.3)$$

$$= \arg \max_{\theta_{i \rightarrow x}} \left\{ \sum_{n_i=0}^{N_i-1} \sum_{t=0}^{M^{(n_i)}-1} \log P_x(x_t^{(n_i)} | x_{0:t-1}^{(n_i)}, i^{(n_i)}; \theta_{i \rightarrow x}) \right\} \quad (6.4)$$

where N_i is the number of image-caption pairs, $M^{(n_i)}$ is the length of the caption $x^{(n_i)}$, x_t denotes a word in the caption, and $P_x(x_t^{(n_i)} | x_{0:t-1}^{(n_i)}, i^{(n_i)})$ corresponds to the activation of the Softmax layer. The decoded word is drawn from:

$$x_t \sim \arg \max_{\mathcal{V}_{i \rightarrow x}^x} P(x_t | x_{0:t-1}; i) \quad (6.5)$$

where $\mathcal{V}_{i \rightarrow x}^x$ is the vocabulary of words in the image-caption dataset $D_{i,x}$.

Neural Machine Translation. Given a pair of source and target sentences (x, y) , the NMT model $P_y(y|x; \theta_{x \rightarrow y})$ computes the conditional probability:

$$P_y(y|x) = \prod_{t=0}^{N-1} P(y_t | y_{0:t-1}; x_{0:M-1}) \quad (6.6)$$

where M and N are the lengths of the source and target sentences, respectively. The maximum-likelihood training objective of the model can be expressed as:

$$\tilde{\theta}_{x \rightarrow y} = \arg \max_{\theta_{x \rightarrow y}} \{ \mathcal{L}_{x \rightarrow y} \} \quad (6.7)$$

$$= \arg \max_{\theta_{x \rightarrow y}} \left\{ \sum_{n_x=0}^{N_x-1} \sum_{t=0}^{N^{(n_x)}-1} \log P_y(y_t^{(n_x)} | y_{0:t-1}^{(n_x)}; x^{(n_x)}; \theta_{x \rightarrow y}) \right\} \quad (6.8)$$

During inference, we calculate the probability of the next symbol given the source sentence encoding and the decoded target sequence so far, and draw the word from the dictionary according to the maximum probability:

$$y_t \sim \arg \max_{\mathcal{V}_{x \rightarrow y}^y} P(y_t | y_{0:t-1}; x_{0:M-1}) \quad (6.9)$$

where $\mathcal{V}_{x \rightarrow y}^y$ is the vocabulary of the target language in the translation dataset $D_{x,y}$.

Unpaired Image Captioning by Language Pivoting. In the unpaired setting, our goal is to generate a description y in the target language for an image i without any pair information. We assume, there is a second language x called “pivot” for which we have (separate) image-pivot and pivot-target paired datasets. The image-to-target model in the pivot-based setting can be decomposed into two sub-models by treating the pivot sentence as a latent variable:

$$P(y|i; \theta_{i \rightarrow x}, \theta_{x \rightarrow y}) = \sum_x P_x(x|i; \theta_{i \rightarrow x}) P_y(y|x; \theta_{x \rightarrow y}) \quad (6.10)$$

where $P_x(x|i; \theta_{i \rightarrow x})$ and $P_y(y|x; \theta_{x \rightarrow y})$ are the image captioning and NMT models, respectively. Due to the exponential search space in the pivot language, we approximate the captioning process with two steps. The first step translates the image i into a pivot language sentence $\tilde{x}$. Then, the pivot language sentence is translated to a target language sentence $\tilde{y}$. To learn such a pivot-based model, a simple approach is to combine the two loss functions in Eq. (6.4) and (6.8) as follows:

$$\mathcal{J}_{i \rightarrow x, x \rightarrow y} = \mathcal{L}_{i \rightarrow x} + \mathcal{L}_{x \rightarrow y} \quad (6.11)$$

During inference, the decoding decision is given by:

$$\tilde{x} = \arg \max_x \{P_x(x|i; \tilde{\theta}_{i \rightarrow x})\} \quad (6.12)$$

$$\tilde{y} = \arg \max_y \{P_y(y|\tilde{x}; \tilde{\theta}_{x \rightarrow y})\} \quad (6.13)$$

where $\tilde{x}$ is the image description generated from i in the pivot language, $\tilde{y}$ is the translation of $\tilde{x}$, and $\tilde{\theta}_{i \rightarrow x}$ and $\tilde{\theta}_{x \rightarrow y}$ are the learned model parameters.

However, this *pipelined* approach to image caption generation in the target language suffers from couple of key limitations. First, image captioning and machine translation are two different tasks. The image-to-pivot and pivot-to-target models are quite different in terms of vocabulary and parameter space because they are trained on two possibly unrelated datasets. Image captions contain description of objects in a given scene, whereas machine translation data is more generic, in our case containing news event descriptions, movie subtitles, and conversational texts. They are two different domains with differences in writing styles and word distributions. As a result, the captions generated by the pipeline approach may not be similar to human-authored captions. Figure 6.1 distinguishes between the two domains of pivot and target sentences: caption domain and translation domain (see second and third circles). The second limitation is that the errors made by the

image-to-pivot captioning model get propagated to the pivot-to-target translation model.

To overcome the limitations of the pivot-based caption generation, we propose to reduce the discrepancy between the image-to-pivot and pivot-to-target models, and to train them jointly so that they learn better models by interacting with each other during training. Figure 6.2 illustrates our approach. The two models share some common aspects that we can exploit to connect them as we describe below.

FIGURE 6.2: Illustration of our image captioning model with pivot language. The image captioning model first transforms an image into latent pivot sentences, from which our machine translation model generates the target caption.

Connecting Image-to-Pivot and Pivot-to-Target. One way to connect the two models is to share the corresponding embedding matrices by defining a common embedding matrix for the decoder of image-to-pivot and the encoder of pivot-to-target. However, since the caption and translation domains are different, their word embeddings should also be different. Therefore, rather than having a common embedding matrix, we add a regularizer $\mathcal{R}_{i \rightarrow y}$ that attempts to bring the input embeddings of the NMT model close to the output embeddings of image captioning model by minimizing their l_2 distance. Formally,

$$\mathcal{R}_{i \rightarrow y}(\theta_{i \rightarrow x}^{w_x}, \theta_{x \rightarrow y}^{w_x}) = - \sum_{w_x \in \mathcal{V}_{i \rightarrow x}^x \cap \mathcal{V}_{x \rightarrow y}^x} \|\theta_{i \rightarrow x}^{w_x} - \theta_{x \rightarrow y}^{w_x}\|_2 \quad (6.14)$$

where w_x is a word in the pivot language that is shared by the two embedding matrices, and $\theta_{i \rightarrow x}^{w_x} \in \mathbb{R}^d$ denotes the vector representation of w_x in the source-to-pivot model, and $\theta_{x \rightarrow y}^{w_x} \in \mathbb{R}^d$ denotes the vector representation of w_x in the pivot-to-target model. Note that, here we adapt $\theta_{x \rightarrow y}^{w_x}$ towards $\theta_{i \rightarrow x}^{w_x}$, that is, $\theta_{i \rightarrow x}^{w_x}$ is already a learned model and kept fixed during adaptation.

Adapting the encoder embeddings of the NMT model does not guarantee that the decoder of the model will produce caption-like sentences. For this, we need to also adapt the decoder embeddings of the NMT model to the caption data. We first use the target-target parallel corpus $D_{\hat{y}, \hat{y}} = \{(\hat{y}^{(n_{\hat{y}})}, \hat{y}^{(n_{\hat{y}})})\}_{n_{\hat{y}}=0}^{N_{\hat{y}}-1}$ to train an autoencoder $P(\hat{y}|\hat{y}; \theta_{\hat{y} \rightarrow \hat{y}})$, where $\theta_{\hat{y} \rightarrow \hat{y}}$ are the parameters of the autoencoder. The maximum-likelihood training objective of autoencoder can be expressed as:

$$\tilde{\theta}_{\hat{y} \rightarrow \hat{y}} = \arg \max_{\theta_{\hat{y} \rightarrow \hat{y}}} \{\mathcal{L}_{\hat{y} \rightarrow \hat{y}}\} \quad (6.15)$$

where $\mathcal{L}_{\hat{y} \rightarrow \hat{y}}$ is the cross-entropy (XE) loss. The autoencoder then “teaches” the decoder of the translation model $P(y|x; \theta_{x \rightarrow y})$ to learn similar word representations. This is again achieved by minimizing the l_2 distance between two vectors:

$$\mathcal{R}_{x \rightarrow \hat{y}}(\theta_{x \rightarrow y}^{w_y}, \theta_{\hat{y} \rightarrow \hat{y}}^{w_y}) = - \sum_{w_y \in \mathcal{V}_{x \rightarrow y}^y \cap \mathcal{V}_{\hat{y} \rightarrow \hat{y}}^y} \|\theta_{x \rightarrow y}^{w_y} - \theta_{\hat{y} \rightarrow \hat{y}}^{w_y}\|_2 \quad (6.16)$$

where $\mathcal{V}_{\hat{y} \rightarrow \hat{y}}^y$ is the vocabulary of y in $D_{\hat{y}, \hat{y}}$, and w_y is a word in the target language that is shared by the two embedding matrices. By optimizing Eq. (6.16), we try to make the learned caption share a similar style as the target captions.

Joint Training. In training, our goal is to find a set of source-to-target model parameters that maximizes the training objective:

$$\mathcal{J}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}} = \mathcal{L}_{i \rightarrow x} + \mathcal{L}_{x \rightarrow y} + \mathcal{L}_{\hat{y} \rightarrow \hat{y}} + \lambda \mathcal{R}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}} \quad (6.17)$$

$$\mathcal{R}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}} = \mathcal{R}_{i \rightarrow y}(\theta_{i \rightarrow x}^{w_x}, \theta_{x \rightarrow y}^{w_x}) + \mathcal{R}_{x \rightarrow \hat{y}}(\theta_{x \rightarrow y}^{w_y}, \theta_{\hat{y} \rightarrow \hat{y}}^{w_y}) \quad (6.18)$$

TABLE 6.1: Statistics of the datasets used in our experiments, where “im” denotes the image, “zh” denotes Chinese, and “en” denotes English.

	Dataset	Lang.	Source		Target	
			# Img/Sent.	# Vocab.	# Sent.	#Vocab.
Training	AIC-ICC	im $\rightarrow$ zh	240K	—	1,200K	4,461
	AIC-MT	zh $\rightarrow$ en	10,000K	50,004	10,000K	50,004
Testing	MSCOCO	im $\rightarrow$ en	123K	—	615K	9,487
	Flickr30K	im $\rightarrow$ en	30K	—	150K	7,000

where λ is the hyper-parameter used to balance the preference between the loss terms and the connection terms. Since both the captioner $P_x(x|i; \theta_{i \rightarrow x})$ and the translator $P_y(y|x; \theta_{x \rightarrow y})$ have large vocabulary sizes (see Table 6.1), it is hard to train the joint model with an initial random policy. Thus, in practice, we pre-train the captioner, translator and autoencoder first, and then jointly optimize them with Eq. (6.17).

6.3 Experiments

Datasets. In our experiments, we choose the two independent datasets used from AI Challenger (AIC) [17]: AIC Image Chinese Captioning (AIC-ICC) and AIC Chinese-English Machine Translation (AIC-MT), as the training datasets, while using MSCOCO and Flickr30K English captioning datasets as the test datasets. Table 6.1 shows the statistics of the datasets used in our experiments.

Training Datasets. For image-to-Chinese captioning training, we follow the settings in AIC-ICC [17] and take 210,000 images for training and 30,000 images for model validation. Each image contains five reference Chinese captions, and each of the captions contains most of the common daily scenes in which a person usually appears. We use the “*Jieba*”¹, a Chinese text segmentation module, for word

¹<https://github.com/fxsjy/jieba>

segmentation. We truncate all the captions longer than 16 tokens and prune the vocabulary by dropping words with a frequency less than 5, resulting in a vocabulary size of 4,461 words.

We take Chinese as the pivot language and learn a Chinese-to-English translation model on AIC-MT. AIC-MT consists of 10,000K Chinese-English parallel sentences, and the English sentences are extracted from English learning websites and movie subtitles. We reserve 4K sentence pairs for validation and 4K sentence pairs for testing. During preprocessing, we remove the empty lines and retain sentence pairs with no more than 50 words. We prune the vocabulary and end up with a vocabulary of size 50,004 words, including special Begin-of-Sentence (BOS) and End-of-Sentence (EOS) tokens. To guide the target decoder to generate caption-like sentences, an autoencoder is also trained with the target image descriptions extracted from MSCOCO. In our training, we extract 60,000 image descriptions from the MSCOCO training split, and randomly sort these samples.

Validation Datasets. We validate the effectiveness of our method on MSCOCO and Flickr30K datasets. The images in MSCOCO typically contain multiple objects with significant contextual information. Likewise, each image in MSCOCO also has five reference description, and most of these descriptions are depicting humans participating in various activities. We use the same test splits as that in [25]. For MSCOCO, we use 5,000 images for validation and 5,000 images for testing, and for Flickr30K, we use 1,000 images for testing.

6.3.1 Implementation Details

Architecture. As can be seen in Figure 6.2, we have three models used in our image captioner. The first model $i2t_{im \rightarrow zh}$ learns to generate the Chinese caption x from a given image i . It is a standard CNN-RNN architecture [112], where the word outputted from the previous time step is taken as the input for the current time

step. For each image, we encode it with ResNet-101 [41], and then apply average pooling to get a vector of dimensions 2,048. After that, we map the image features through a linear projection and get a vector of dimensions 512. The decoder is implemented based on an LSTM network. The dimensions of the LSTM hidden states and word embedding are fixed to 512 for all of the models discussed in this work. Each sentence starts with a special BOS token and ends with an EOS token.

The second model $\text{nmt}_{\text{zh} \rightarrow \text{en}}$ learns to translate the Chinese sentence x to the English sentence y . It has three components: a sentence encoder, a sentence decoder, and an attention module. The words in the pivot language are first mapped to word vectors and then fed into a bidirectional LSTM network. The decoder predicts the target language words based on the encoded vector of the source sentence as well as its previous outputs. The encoder and the decoder are connected through an attention module, which allows the decoder to focus on different regions of the source sentence during decoding.

The third model $\text{t2t}_{\text{en} \rightarrow \text{en}}$ learns to produce the caption-style English sentence $\hat{y}$. It is essentially an autoencoder trained on a set of image descriptions extracted from MSCOCO, where the encoder and the decoder are based on one-layer LSTM network. The encoder reads the whole sentence as input, and the decoder is to reconstruct the input sentence.

Training Setting. All the modules are randomly initialized before training except the image CNN, for which we use a pre-trained model on ImageNet. We first independently train the image Chinese captioner, the Chinese-to-English translator, and the autoencoder with the cross-entropy loss on AIC-ICC, AIC-MT, and MSCOCO corpus, respectively. During this stage, we use Adam [110] algorithm to do model updating with a mini-batch size of 100. The initial learning rate is $4e^{-4}$, and the momentum is 0.9. The best models are selected according to the validation scores, which are then used for the subsequent joint training. Specifically, we combine the just trained models with the connection terms, and conduct

a joint training with Eq. (6.17). We set the hyper-parameter λ to 1.0, and train the joint model using Adam optimizer with a mini-batch size of 64 and an initial learning rate of $2e^{-4}$. Weight decay and dropout are applied in this training phase to prevent over-fitting.

Testing setting. During testing, the output image description is first formed by drawing words in pivot language from $i2t_{im \rightarrow zh}$ until an EOS token is reached, and then translated with $nmt_{zh \rightarrow en}$ to the target language. Here we use beam search for the two inference procedures. Beam search is an efficient decoding method for RNN-based models, which keeps the top- k hypotheses at each time step, and considers them as the candidates to generate a new top- k hypotheses at the next time step. We set a fixed beam search size of $k = 5$ for $i2t_{im \rightarrow zh}$ and $k = 10$ for $nmt_{zh \rightarrow en}$. We evaluate the quality of the generated image descriptions with the standard evaluation metrics: BLEU [65], METEOR [108], and CIDEr [66]. Since BLEU aims to assess how similar two sentences are, we also evaluate the diversity of the generated sentence with Self-BLEU [135], which takes one sentence as the hypothesis and the others as the reference, and then calculates BLEU score for every generated sentence. The final Self-BLEU score is defined as the average BLEU scores of the sentences.

6.3.2 Quantitative Analysis

Results of Image Chinese Captioning. Table 6.2 shows the comparison results on the AIC-ICC validation set, where B@n is short for BLEU-n. We compare our $i2t_{im \rightarrow zh}$ model with the baseline [17] (named AIC-I2T). Both AIC-I2T and our image caption model ($i2t_{im \rightarrow zh}$) are trained with cross-entropy loss. We can see that our model outperforms the baseline in all the metrics. This might be due to different implementation details, *e.g.*, AIC-I2T utilizes Inception-v3 for the image CNN while we use ResNet-101.

TABLE 6.2: Performance comparisons on AIC-ICC. The results of $i2t_{im \rightarrow zh}$ are achieved via beam search.

Image Chinese Captioning					
Approach	B@1	B@2	B@3	B@4	CIDEr
$i2t_{im \rightarrow zh}$	77.8	65.9	55.5	46.6	144.2
AIC-I2T [17]	76.5	64.8	54.7	46.1	142.5

TABLE 6.3: Performance comparisons on AIC-MT test dataset. Note that our $nmt_{zh \rightarrow en}$ model uses beam search.

Chinese-to-English Translation		
Approach	Accuracy	Perplexity
$nmt_{zh \rightarrow en}$	55.0	8.9
Google Translation	57.8	—

Results of Chinese-to-English Translation. Table 6.3 provides the comparison between our attention-based machine translator with the online Google translator on AIC-MT test split. We use the “googletrans”², a free Python tool that provides Google translator API. The perplexity value in the second column is the geometric mean of the inverse probability for each predicted word. Our attention-based NMT model ($nmt_{zh \rightarrow en}$) is trained on AIC-MT training set. We can see that our model is slightly worse than online Google translation in accuracy. This not surprising considering that Google’s translator is trained on much larger datasets with more vocabulary coverage, and it is a more complex system that ensembles multiple NMT models.

Results of Unpaired Image English Captioning. Table 6.4 shows the comparisons among different variants of our method on MSCOCO dataset. Our upper bound is achieved by an image captioning model $i2t_{im \rightarrow en}$ that is trained with paired English captions. $i2t_{im \rightarrow en}$ shares the same architecture as $i2t_{im \rightarrow zh}$, except that they have different vocabulary sizes. The lower bound is achieved by pipelining $i2t_{im \rightarrow zh}$ and $nmt_{zh \rightarrow en}$. In the pipeline setting, these two models are trained on AIC-ICC and AIC-MT, respectively. We also report the results of our implementation of FC-2K [28], which adopts a similar architecture.

²<https://pypi.python.org/pypi/googletrans>

TABLE 6.4: Results of unpaired image-to-English captioning on MSCOCO 5K and Flickr30K 1K test splits, where M is short for METEOR.

Approach	Lang.	B@1	B@2	B@3	B@4	M	CIDEr
<i>MSCOCO</i>							
i2t _{im→en} (Upper bound, XE Loss)	en	73.2	56.3	42.0	31.2	25.3	95.1
FC-2K [28] (ResNet101, XE Loss)	en	–	–	–	29.6	25.2	94.0
i2t _{im→zh} + nmt _{zh→en} ($\mathcal{R}_{i→x,x→y,y→\hat{y}}$)	en	46.2	24.0	11.2	5.4	13.2	17.7
i2t _{im→zh} + nmt _{zh→en} ($\mathcal{R}_{i→x,x→y}$)	en	45.5	23.6	11.0	5.3	13.1	17.3
i2t _{im→zh} + nmt _{zh→en} (Lower bound)	en	42.0	20.6	9.5	3.9	12.0	12.3
i2t _{im→zh} + Online Google Translation	en	42.2	21.8	10.7	5.3	14.5	17.0
<i>Flickr30K</i>							
i2t _{im→en} (Upper bound, XE Loss)	en	63.1	43.8	30.2	20.7	17.7	40.1
i2t _{im→zh} + nmt _{zh→en} ($\mathcal{R}_{i→x,x→y,y→\hat{y}}$)	en	49.7	27.8	14.8	7.9	13.6	16.2
i2t _{im→zh} + nmt _{zh→en} ($\mathcal{R}_{i→x,x→y}$)	en	48.7	26.1	12.8	6.4	13.0	14.9
i2t _{im→zh} + nmt _{zh→en} (Lower bound)	en	45.9	25.2	13.1	6.9	12.5	13.9
i2t _{im→zh} + Online Google Translation	en	46.2	25.4	13.9	7.7	14.4	15.8

For unpaired image-to-English captioning, our method with the connection term on pivot language ($\mathcal{R}_{i→x,x→y}^{w_x}$) outperforms the method of combining i2t_{im→zh} with on-line Google translation in terms of B@n and CIDEr metrics, while obtaining significant improvements over the lower bound. This demonstrates the effectiveness of the connection term on the pivot language. Moreover, by adding the connection term on the target language, our model with the two connection terms ($\mathcal{R}_{i→x,x→y,y→\hat{y}}$) further improves the performance. This suggests that a small corpus in the target domain is able to make the decoder to generate image descriptions that are more like captions. The connection terms help to bridge the word representations of the two different domains. The captions generated by Google translator have higher METEOR. We speculate the following reasons. First, Google Translator generates longer captions than ours. Since METEOR computes the score not only on the basis of n -gram precision but also of uni-gram recall, its default parameters favor longer translations than other metrics [136]. Second, in addition to exact word matching, METEOR considers the matching of word stems and synonyms. Since Google translator is trained on a much larger corpus than ours, it generates more

TABLE 6.5: Self-BLEU scores on MSCOCO 5K test split. Note that lower Self-BLEU scores imply higher diversity of the image descriptions.

Approach	Lang.	Self-B@2	Self-B@3	Self-B@4	Self-B@5
i2t _{im→en} (GT Captions)	en	85.0	67.8	49.1	34.4
i2t _{im→en} (Upper bound)	en	99.0	97.5	94.6	90.7
i2t _{im→zh} + nmt _{zh→en} ($\mathcal{R}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}}$)	en	95.6	91.7	86.5	80.2
i2t _{im→zh} + nmt _{zh→en} (Lower bound)	en	98.1	95.9	92.3	87.6

synonymous words. Table 6.4 also shows the results of unpaired image English captioning on Flickr30K, where we can draw similar conclusions.

We further evaluate the diversity of the generated image descriptions using the Self-BLEU metric. Table 6.5 shows the detailed Self-BLEU scores. It can be seen that our method generates image descriptions with the highest diversity, compared with the upper and lower bounds. For better comparison, we also calculate the Self-BLEU scores calculated on ground truth captions.

We also conduct a human evaluation of the generated captions for different models as well as the ground truth captions. A total number of 12 evaluators of different educational background were invited, and a total of 1.2K samples were randomly selected from the test split for the user study. Particularly, we measure the caption quality from two aspects: *relevant* and *resemble*. The *relevant* metric indicates whether the caption is correct according to the image content. The *resemble* metric assesses to what extent the systems produce captions that resemble human-authored captions. The evaluators assessed the quality in 5 grades: 1-very poor, 2-poor, 3-barely acceptable, 4-good, 5-very good. Each evaluator assessed randomly chosen 100 images. The results presented in Table 6.6 demonstrate that our approach can generate relevant and human understandable image captions as the paired (Upper bound) approach.

TABLE 6.6: Evaluation results of user assessment on MSCOCO 1.2K test split.

Approach	$i2t_{im \rightarrow zh} + nmt_{zh \rightarrow en}(\mathcal{R}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}})$	Upper Bound	Ground-Truth
Relevant	3.81	3.99	4.68
Resemble	3.78	4.05	4.48

6.3.3 Qualitative Results

We provide some captioning examples in Figure 6.3 for a better understanding of our model. We show in different color the generated captions for several images by the three models along with the ground truth (GT) captions. From these exemplary results, we can see that, compared with the paired model $i2t_{im \rightarrow en}$, our pivot-based unpaired model $i2t_{im \rightarrow zh \rightarrow en}(\mathcal{R}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}})$ often generates more diverse captions; thanks to the additional translation data. At the same time, our model can generate caption-like sentences by bridging the gap between the datasets and by joint training of the model components. For example, with the detected people in the first image, our model generates the sentence with “*a bunch of people in sports suits*”, which is more diverse than the sentence with “*a group of baseball players*” generated by the paired model.

FIGURE 6.3: Examples of the generated sentences on MSCOCO test images, where $i2t_{im \rightarrow zh}$ is the image captioner trained on AIC-ICC, $i2t_{im \rightarrow en}$ is the image captioner trained on MSCOCO, $i2t_{im \rightarrow zh \rightarrow en}(\mathcal{R}_{i \rightarrow x, x \rightarrow y, y \rightarrow \hat{y}})$ and $i2t_{im \rightarrow zh \rightarrow en}(\mathcal{R}_{i \rightarrow x, x \rightarrow y})$ are our proposed models for unpaired image captioning, and GT stands for ground truth caption.

6.4 Summary

In this chapter, we have proposed an approach to unpaired image captioning with the help of a pivot language. Our method couples an image-to-pivot captioning model with a pivot-to-target NMT model in a joint learning framework. The coupling is done by adapting the word representations in the encoder and the decoder of the NMT model to produce caption-like sentences. Empirical evaluation demonstrates that our method consistently outperforms the baseline methods on MSCOCO and Flickr30K image captioning datasets.

Chapter 7

Unpaired Image Captioning via Scene Graph Alignment

7.1 Introduction

Today’s image captioning models heavily depend on paired image-caption datasets. Most of them employ an encoder-decoder framework [11, 13, 19, 112, 137], which uses a CNN [5] to encode an image into a feature vector and then a RNN to decode it into a text sequence. However, it is worthwhile noticing that the overwhelming majority of image captioning studies are conducted in English [138]. The bottleneck is the lack of large scale image-caption paired datasets in other languages, and getting such paired data for each target language requires human expertise in a time-consuming and labor-intensive process.

Several encoder-decoder models have been proposed in recent years for unsupervised neural machine translation [29, 139]. The key idea of these methods mainly relies on training denoising auto-encoders for language modeling and on sharing latent representations across the source and target languages for the encoder and

FIGURE 7.1: Illustration of our graph-based learning method. Our model consists of one visual scene graph detector (Top-Left), one fixed off-the-shelf scene graph language parser (Bottom-Left), a scene graph encoder G_{Enc}^S , a sentence decoder G_{Dec}^S , and a feature mapping module.

the decoder. Despite the promising results achieved by the unsupervised neural machine translation, unpaired image-to-sentence translation is far from mature.

Recently, there have been few attempts at relaxing the requirement of paired image-caption data for this task. The first work in this direction is the pivot-based semi-supervised solution proposed by Gu *et al.* [12], where they take a pivot language as a bridge to connect the source image and the target language caption. Their method requires an image-text paired data for the pivot language (Chinese), and a parallel corpus for the pivot to target translation. Feng *et al.* [80] move a step further, where they conduct purely unsupervised image captioning without relying on any labeled image-caption pairs. Their method uses a sentence discriminator along with a visual concept detector to connect the image and the text modalities through adversarial training. Although promising, the results of the existing methods are still far below compared to their paired counterparts.

Unlike unsupervised neural machine translation, where the encoders can be shared across the source and target languages, due to the different structures and characteristics of image and text modalities, the encoders of image and sentence cannot

be shared to connect the two modalities. The critical challenge in unpaired image captioning is, therefore, the gap of information misalignment in images and sentences, so as to fit the encoder-decoder framework.

Fortunately, with recent breakthroughs in deep learning and image recognition, higher-level visual understanding tasks such as scene graph construction have become popular research topics with significant advancements [14, 122, 140–144]. Scene graph, as an abstraction of objects and their complex relationships, provide rich semantic information of an image. The value of scene graph representation has been proven in a wide range of vision-language tasks, such as visual question answering [145] and paired image captioning [144].

Considering the significant challenges that unpaired image captioning problem poses in terms of different characteristics between visual and textual modalities, in this section, we propose a scene graph-based method that exploits the rich semantic information captured by scene graphs. Our framework comprises an image scene graph generator, a sentence scene graph generator, a scene graph encoder, a sentence decoder, and a feature alignment module that maps the features from image to sentence modality. Figure 7.1 sketches our solution. We first extract the sentence scene graphs from the sentence corpus and train the scene graph encoder and the sentence decoder on the text modality. To align the scene graphs between images and sentences, we use CycleGAN [30] to build the data correspondence between the two modalities. Specifically, given the unrelated image and sentence scene graphs, we first encode them with the scene graph encoder trained on the sentence corpus. Then, we perform unsupervised cross-modal mapping for feature level alignments with CycleGAN. By mapping the features, the encoded image scene graph is pushed close to the sentence modality, which is then used effectively as input to the sentence decoder to generate meaningful sentences.

The main contributions of this work include: (1) a novel scene graph-based framework for unpaired image captioning; (2) an unsupervised feature alignment method

that learns the cross-modal mapping without any paired data. Our experimental results demonstrate the effectiveness of our proposed model in producing quite promising image captions. The comparison with recent unpaired image captioning methods validates the superiority of our method.

7.2 Method

In this section, we describe our unpaired image captioning framework. We first revisit the paired captioning setting.

7.2.1 Paired Image Captioning Revisited

In the paired captioning setting, our training goal is to generate a caption S from an image I such that S is similar to its ground-truth caption. The popular encoder-decoder framework for image captioning can be formulated as:

$$P(S|I) = \underbrace{P(V|I)}_{\text{Encoder}} \underbrace{P(S|V)}_{\text{Decoder}} \quad (7.1)$$

where the encoder $P(V|I)$ encodes the image I into the image features V with a CNN model [5], and the decoder $P(S|V)$ predicts the image description S from the image features V . The most common training objective is to maximize the probability of the ground-truth caption words given the image: $\sum_t \log p_{\theta_{I \rightarrow S}}(S_t|S_{0:t-1}, I)$, where $p_{\theta_{I \rightarrow S}}(S_t|S_{0:t-1}, I)$ corresponds to the Softmax output at time step t . During inference, the word S_t is drawn from the dictionary $\mathcal{D}_S$ according to the Softmax distribution.

7.2.2 Unpaired Image Captioning

In the unpaired image captioning setting, we have a dataset of images $\mathcal{I} = \{I_1, \dots, I_{N_I}\}$, and a dataset of sentences $\mathcal{S} = \{S_1, \dots, S_{N_S}\}$, where N_I and N_S are the total numbers of images and sentences, respectively. In this setting, there is no alignment between $\mathcal{I}$ and $\mathcal{S}$. In fact, $\mathcal{I}$ and $\mathcal{S}$ can be completely unrelated coming from two different domains. Our goal is to train an image captioning model in a completely unsupervised way. In our setup, we assume that we have access to an off-the-shelf image scene graph detector and a sentence (or text) scene graph parser.

As shown in Figure 7.1, our proposed image captioning model consists of one image scene graph generator, one sentence scene graph generator, one scene graph encoder G_{Enc}^S , one attention-based decoder for sentence generation G_{Dec}^S , and a cycle-consistent feature alignment module. Given an image I as input, our method first extracts an image scene graph $\mathcal{G}^I$ using the scene graph generator. It then maps $\mathcal{G}^I$ to the sentence scene graph $\mathcal{G}^S$ from which the RNN-based decoder generates a sentence S . More formally, the image captioner $P(S|I)$ in the unpaired setting can be decomposed into the following submodels,

$$I \rightarrow \mathcal{G}^I \tag{7.2}$$

$$P(S|I) = \underbrace{P(\mathcal{G}^S|\mathcal{G}^I)}_{\text{Unpaired Mapping}} \underbrace{P(S|\mathcal{G}^S)}_{\text{Decoder}} \tag{7.3}$$

where $\mathcal{G}^I$ and $\mathcal{G}^S$ are the image scene graph and the sentence scene graph, respectively. The most crucial component in Eq. (7.3) is the unpaired mapping of image and text scene graphs. In our approach, this mapping is done in the feature space. In particular, we encode the image and sentence scene graphs into feature vectors and learn to map the feature vectors across the two modalities. We reformulate

Eq. (7.3) as follows:

$$\begin{aligned} P(S|I) &= P(\mathcal{G}^S|\mathcal{G}^I)P(S|\mathcal{G}^S) \\ &\approx P(\mathbf{f}^I|\mathcal{G}^I)P(\mathbf{f}^S|\mathbf{f}^I)P(S|\mathbf{f}^S) \end{aligned} \quad (7.4)$$

where $P(\mathbf{f}^I|\mathcal{G}^I)$ is a graph encoder, $P(S|\mathbf{f}^S)$ is an RNN-based sentence decoder, and $P(\mathbf{f}^S|\mathbf{f}^I)$ is a cross-modal feature mapper in the unpaired setting. In our implementation, we learn the scene graph encoder and the RNN-based decoder on the text modality first, and then we try to map the image scene graph into a common feature space (*i.e.*, the text space) so that the same sentence decoder can be used to decode the sentence from the mapped image features.

The sentence encoding and decoding processes can be formulated as the following two steps:

$$S \rightarrow \mathcal{G}^S \quad (7.5)$$

$$\hat{S} = \arg \max_S P(S|\mathbf{f}^S)P(\mathbf{f}^S|\mathcal{G}^S) \quad (7.6)$$

where $\hat{S}$ is the reconstructed sentence. We train the model to enforce $\hat{S}$ to be close to the original sentence S .

In the following, we describe the scene graph generator in Sec. 7.2.2.1, the scene graph encoder in Sec. 7.2.2.2, the sentence decoder in Sec. 7.2.2.3, and our unpaired feature mapping process in Sec. 7.2.2.4.

7.2.2.1 Scene Graph Generator

Formally, a scene graph is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ containing a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. As exemplified in Figure 7.1, the nodes can be of three types: object node, attribute node, and relationship node. We denote o_i as the i -th object, $r_{i,j}$ as the relation between object o_i and o_j , and a_i^l as the l -th attribute of object o_i .

An image scene graph generator contains an object detector, an attribute classifier, and a relationship classifier. We use Faster-RCNN [146] as the object detector, MOTIFS [147] as the relationship detector, and an additional classifier for attribute identification [144].

To generate the sentence scene graph $\mathcal{G}^S$ for a sentence, we first parse the sentence into a syntactic tree using the parser provided by [109], which uses a syntactic dependency tree built by [148]. Then, we transform the tree into a scene graph with a rule-based method [149].

7.2.2.2 Scene Graph Encoder

We follow [144] to encode a scene graph. Specifically, we represent each node as a d_e -dimensional feature vector, and use three different spatial graph convolutional encoders to encode the three kinds of nodes by considering their neighborhood information in the scene graph.

Encoding objects. In a scene graph (image or sentence), an object o_i can play either a *subject* or an *object* role in a relation triplet depending on the direction of the edge. Therefore, for encoding objects, we consider what relations they are associated with and what roles they play in that relation. Let $\langle o_i, o_j, r_{i,j} \rangle$ denote the triplet for relation $r_{i,j}$, where o_i plays a *subject* role and o_j plays an *object* role. The encoding for object o_i , that is $\mathbf{x}_{o_i} \in \mathbb{R}^{d_x}$ is computed by:

$$\mathbf{x}_{o_i} = \frac{1}{N_{r_i}} \sum_{o_j} g_s(\mathbf{e}_{o_i}, \mathbf{e}_{o_j}, \mathbf{e}_{r_{i,j}}) + \frac{1}{N_{r_i}} \sum_{o_k} g_o(\mathbf{e}_{o_k}, \mathbf{e}_{o_i}, \mathbf{e}_{r_{k,i}}) \quad (7.7)$$

where $\mathbf{e}_{o_i} \in \mathbb{R}^{d_e}$ and $\mathbf{e}_{r_{i,j}} \in \mathbb{R}^{d_e}$ are the embeddings (randomly initialized) representing the object o_i and the relation $r_{i,j}$, respectively; $g_s(\cdot)$ and $g_o(\cdot)$ are the

spatial graph convolution operations for objects as a *subject* and as an *object*, respectively; and N_{r_i} is the total number of relation triplets that o_i is associated with in the scene graph.

Encoding attributes. An object o_i may have multiple attributes in the scene graph. The encoding of an object based on its attributes, *i.e.*, $\mathbf{x}_{a_i} \in \mathbb{R}^{d_x}$ is computed by:

$$\mathbf{x}_{a_i} = \frac{1}{N_{a_i}} \sum_l g_a(\mathbf{e}_{o_i}, \mathbf{e}_{a_i^l}) \quad (7.8)$$

where N_{a_i} is the total number of attributes that object o_i has, and $g_a(\cdot)$ is the spatial convolutional operation for attribute based encoding.

Encoding relations. Each relation $r_{i,j}$ is encoded into $\mathbf{x}_{r_{i,j}} \in \mathbb{R}^{d_x}$ by considering the objects that the relation connects in the relation triplet,

$$\mathbf{x}_{r_{i,j}} = g_r(\mathbf{e}_{o_i}, \mathbf{e}_{o_j}, \mathbf{e}_{r_{i,j}}) \quad (7.9)$$

where $g_r(\cdot)$ is the associated convolutional operation.

After graph encoding, for each image scene graph or sentence scene graph, we have three sets of embeddings:

$$\begin{aligned} \mathcal{X}_o^k &= [\mathbf{x}_{o_1}^k, \dots, \mathbf{x}_{o_{N_o^k}}^k] \\ \mathcal{X}_r^k &= [\mathbf{x}_{r_1}^k, \dots, \mathbf{x}_{r_{N_r^k}}^k], \quad k \in \{I, S\} \\ \mathcal{X}_a^k &= [\mathbf{x}_{a_1}^k, \dots, \mathbf{x}_{a_{N_a^k}}^k] \end{aligned} \quad (7.10)$$

where N_o^k , N_r^k , and N_a^k can be different from each other. Figure 7.2 illustrates the encoding process.

FIGURE 7.2: The architectures for scene graph encoding, attention, and sentence decoding; g_{Att}^o , g_{Att}^r , and g_{Att}^a are attention modules for each kind of features, respectively.

7.2.2.3 Sentence Decoder

The goal of the sentence decoder is to generate a sentence $\hat{S}$ from the encoded embeddings, $\mathcal{X}_o^k$, $\mathcal{X}_r^k$, and $\mathcal{X}_a^k$. However, these three sets of embeddings are of different lengths and contain different information. Therefore, their importance for the sentence decoding task also vary. In order to compute a relevant context for the decoding task effectively, we use three attention modules, one for each type of embeddings. The attention module g_{Att}^o over $\mathcal{X}_o^k$ is defined as:

$$\mathbf{f}_o^k = \sum_{i=1}^{N_o^k} \alpha_i \mathbf{x}_{o_i}^k; \quad \alpha_i = \frac{\exp(\mathbf{w}_o^T \mathbf{x}_{o_i}^k)}{\sum_m \exp(\mathbf{w}_o^T \mathbf{x}_{o_m}^k)} \quad (7.11)$$

where $\mathbf{w}_o$ is the associated (learnable) weight vector. The attentions over $\mathcal{X}_r^k$, and $\mathcal{X}_a^k$ are similarly defined to get the respective attention vectors, $\mathbf{f}_r^k \in \mathbb{R}^{d_f}$ and $\mathbf{f}_a^k \in \mathbb{R}^{d_f}$.

The attention vectors are then combined to get a triplet level embedding, which is then fed into an RNN-based decoder to generate the sentence $\hat{S}$. The following

sequence of operations formally describes the process.

$$\mathbf{f}_{\text{ora}}^k = g_{\text{ora}}([\mathbf{f}_o^k, \mathbf{f}_r^k, \mathbf{f}_a^k]) \quad (7.12)$$

$$\mathbf{o}_t, \mathbf{h}_t = \text{RNN}_{\text{Dec}}(\mathbf{f}_{\text{ora}}^k, \mathbf{h}_{t-1}, \hat{S}_{t-1}) \quad (7.13)$$

$$\hat{S}_t \approx \text{Softmax}(\mathbf{W}_o \mathbf{o}_t) \quad (7.14)$$

where $g_{\text{ora}}(\cdot)$ is a neural network that generates a triplet level embedding, and $\mathbf{o}_t$ is the cell output of the decoder at time step t .

7.2.2.4 Training and Inference

We first train the graph encoder and the sentence decoder in the text modality (Eq. (7.6)), and then perform a feature level alignments for cross-modal unsupervised mapping.

Training in Text Modality. The graph convolutional encoding of a sentence scene graph $\mathcal{G}^S$ into a feature representation $\mathbf{f}_{\text{ora}}^S$, and reconstructing the original sentence S from it are shown at the bottom part of Figure 7.1, where the encoder and the decoder are denoted as G_{Enc}^S and G_{Dec}^S , respectively. We first train G_{Enc}^S and G_{Dec}^S models by minimizing the cross-entropy (XE) loss:

$$\mathcal{L}_{\text{XE}}(\theta_{\mathcal{G} \rightarrow S}) = - \sum_t \log p_{\theta_{\mathcal{G} \rightarrow S}}(S_t | S_{0:t-1}) \quad (7.15)$$

where $\theta_{\mathcal{G} \rightarrow S}$ are the parameters of G_{Enc}^S and G_{Dec}^S , $p_{\theta_{\mathcal{G} \rightarrow S}}(S_t | S_{0:t-1})$ is the output probability of t -th word in the sentence given by the sentence decoder.

We further employ a reinforcement learning (RL) loss that takes the entire sequence into account. Specifically, we take the CIDEr [66] score as the reward and optimize $\theta_{\mathcal{G} \rightarrow S}$ by minimizing the negative expected rewards as follows:

$$\mathcal{L}_{\text{RL}}(\theta_{\mathcal{G} \rightarrow S}) = -\mathbb{E}_{\tilde{S} \sim P_{\theta_{\mathcal{G} \rightarrow S}}} [r(\tilde{S})] \quad (7.16)$$

FIGURE 7.3: Conceptual illustration of our unpaired feature mapping. For each kind of embedding p , there are two mapping functions $F_{I \rightarrow S}^p$ and $F_{S \rightarrow I}^p$, and two associated adversarial discriminators D_I^p and D_S^p .

where $r(\tilde{S})$ is the reward calculated by comparing the sampled sentence $\tilde{S}$ with the ground-truth sentence S using the CIDEr metric. In our model, we follow the RL approach proposed in [11, 28].

Unsupervised Mapping of Scene Graph Features. To adapt the learned model from sentence modality to the image modality, we need to translate the scene graph from the image to the sentence modality. We take the discrepancy in the modality of scene graphs directly into account by aligning the representation of the image scene graph with the sentence scene graph. We propose to use CycleGAN [30] to learn the feature alignment across domains.

Figure 7.3 illustrates our idea. Given two sets of unpaired features $\mathbf{f}_p^I$ and $\mathbf{f}_p^S$, where $p \in \{o, r, a\}$, we have two mapping functions $F_{I \rightarrow S}^p(\cdot)$ and $F_{S \rightarrow I}^p(\cdot)$, and two discriminators D_S^p and D_I^p . $F_{I \rightarrow S}^p(\cdot)$ maps the image features to the sentence features, and $F_{S \rightarrow I}^p(\cdot)$ maps the sentence features to the image features. The discriminators are trained to distinguish the *real* (original modality) features from the *fake* (mapped) features. The mappers are trained to fool the respective discriminators through adversarial training.

For image to text mapping, the adversarial loss is:

$$\mathcal{L}_{\text{GAN}}(F_{I \rightarrow S}^p, D_S^p) = \mathbb{E}_S[\log D_S^p(\mathbf{f}_{p,\text{Real}}^S)] + \mathbb{E}_I[\log(1 - D_S^p(F_{I \rightarrow S}^p(\mathbf{f}_{p,\text{Real}}^I))] \quad (7.17)$$

Similarly, for sentence to image mapping, we have the similar adversarial loss,

$$\mathcal{L}_{\text{GAN}}(F_{S \rightarrow I}^p, D_I^p) = \mathbb{E}_I[\log D_I^p(\mathbf{f}_{p,\text{Real}}^I)] + \mathbb{E}_S[\log(1 - D_I^p(F_{S \rightarrow I}^p(\mathbf{f}_{p,\text{Real}}^S))] \quad (7.18)$$

Due to the unpaired setting, the mapping from the source to the target modality is highly under-constrained. To make the mapping functions cycle-consistent, CycleGAN introduces a cycle consistency loss to regularize the training,

$$\mathcal{L}_{\text{cyc}}(F_{S \rightarrow I}^p, F_{I \rightarrow S}^p) = \mathbb{E}_I[\|\mathbf{f}_{p,\text{Rec}}^I - \mathbf{f}_{p,\text{Real}}^I\|_1] + \mathbb{E}_S[\|\mathbf{f}_{p,\text{Rec}}^S - \mathbf{f}_{p,\text{Real}}^S\|_1] \quad (7.19)$$

where $\mathbf{f}_{p,\text{Rec}}^I$ and $\mathbf{f}_{p,\text{Rec}}^S$ are the reconstructed features in the image and text modalities, respectively.

Formally, our overall objective for unpaired feature mapping is to optimize the following loss:

$$\mathcal{L}_{\text{Adv}}(\theta_{I \leftrightarrow S}) = \sum_{p \in \{o, r, a\}} \mathcal{L}_{\text{Adv}}(\theta_{I \leftrightarrow S}^p) \quad (7.20)$$

$$\begin{aligned} \mathcal{L}_{\text{Adv}}(\theta_{I \leftrightarrow S}^p) &= \mathcal{L}_{\text{GAN}}(F_{S \rightarrow I}^p, D_I^p) + \mathcal{L}_{\text{GAN}}(F_{I \rightarrow S}^p, D_S^p) \\ &\quad + \lambda \mathcal{L}_{\text{cyc}}(F_{S \rightarrow I}^p, F_{I \rightarrow S}^p) \end{aligned} \quad (7.21)$$

where $\theta_{I \leftrightarrow S}^p$ are the parameters of the two mapping functions and the discriminators for each kind of embedding p , and λ is a hyperparameter to control the regularization.

Cross-modal Inference. During inference, given an image I , we generate its corresponding scene graph $\mathcal{G}^I$ using a pre-trained image scene graph generator,

use the scene graph encoder to get the image features $\mathbf{f}_p^I$, which are then mapped through the image-to-text mapper $F_{I \rightarrow S}^p$. The mapped features are then used for sentence generation using the sentence decoder. The cross-modal inference process can be formally expressed as:

$$\hat{S} = \arg \max_S P(S | \mathbf{f}_{\text{ora}}^I) P(\mathbf{f}_r^I, \mathbf{f}_o^I, \mathbf{f}_a^I | \mathcal{G}^I) \quad (7.22)$$

$$\mathbf{f}_{\text{ora}}^I = g_{\text{ora}}([F_{I \rightarrow S}^o(\mathbf{f}_o^I), F_{I \rightarrow S}^r(\mathbf{f}_r^I), F_{I \rightarrow S}^a(\mathbf{f}_a^I)]) \quad (7.23)$$

where $g_{\text{ora}}(\cdot)$ is the same module as Eq. (7.12).

7.3 Experiments

In this section, we evaluate the effectiveness of our proposed method. We first introduce the datasets and the experimental settings. Then, we present the performance comparisons as well as ablation studies to understand the impact of different components of our framework.

7.3.1 Datasets and Setting

Table 7.1 shows the statistics of the training datasets used in our experiments. We use Visual Genome (VG) dataset [150] to train our image scene graph generator. We filter the object, attribute, and relation annotations by keeping those that appear more than 2,000 times in the training set. The resulting dataset contains 305 objects, 103 attributes, and 64 relations (a total of 472 items).

We collect the image descriptions from the training split of MSCOCO [15] and use them as our sentence corpus to train the scene graph encoder and the sentence decoder. In pre-processing, we tokenize the sentences and convert all the tokens to lowercase. The tokens that appear less than five times are treated as $\langle \text{UNK} \rangle$ tokens.

The maximum caption length is fixed to 16, and all the captions longer than 16 are truncated. This results in a base vocabulary of 9,487 words. For sentence scene graph generation, we generate the scene graph using the language parser in [109, 144]. We perform a filtering process by removing objects, relations, and attributes that appear less than 10 times in all the parsed scene graphs. After this filtering, we obtain 5,364 objects, 1,308 relations, and 3,430 attributes. This gives an extended vocabulary where the previous 9,487 words are consistent with the base vocabulary. The embeddings for the vocabulary items are randomly initialized.

TABLE 7.1: Statistics of the training datasets.

Scene Graph	Vocabulary Size		
	#Object	#Attribute	#Relation
Image (VG)	305	103	64
Sentence (MSCOCO)	5,364	3,430	1,308

For learning the mapping between the modalities, the unpaired training data is intentionally collected by shuffling the images and the sentences from MSCOCO randomly. We validate the effectiveness of our method on the same test splits as used in [12, 80] for a fair comparison. The widely used CIDEr-D [66], BLEU [65], METEOR [151], and SPICE [109] are used to measure the quality of the generated captions.

7.3.2 Implementation Details

We follow [144] to train our image scene graph generator on VG. We first train a Faster-RCNN and use it to identify the objects in each image. We select at least 10 and at most 100 objects for an image. The object features extracted by RoI pooling are used as input to the object detector, the relation classifier, and the attribute classifier. We adopt the LSTM-based relation classifier from [147]. Our attribute classifier is a single hidden layer network with ReLU activation (*i.e.*, fc-ReLU-fc-Softmax), and we keep only the three most probable attributes for each object. For scene graph encoding, we set $d_e = d_x = d_f = 1000$. We implement g_s , g_o , g_r , g_a , and

g_{ora} (Eq. (7.7) - (7.12)) as fully-connected layers with ReLU activations. The two mapping functions in Eq. (7.17) and Eq. (7.18) are implemented as fully-connected layers with leaky ReLU activations.

The sentence decoder has two LSTM layers. The input to the first LSTM is the word embeddings and its previous hidden state. The input to the second LSTM is the concatenation of three terms: the triplet embedding $\mathbf{f}_{\text{ora}}$, the output from the first LSTM, and its previous hidden state. We set the number of hidden units in each LSTM to 1,000.

During training, we first train the network with the cross-entropy loss (Eq. (7.15)) for 20 epochs and then fine-tune it with RL loss in Eq. (7.16). The learning rate is initialized to 4×10^{-4} for all parameters and decayed by 0.8 after every 5 epoch. We use Adam [110] for optimization with a batch size of 50. During the (unpaired) alignment learning, we freeze the parameters of the scene graph encoder and the sentence decoder, and only learn the mapping functions and the discriminators. For all the experiments, we empirically set λ to 10 in Eq. (7.21). During inference, we use beam search with a beam size of 5.

For quantifying the efficacy of the proposed framework, we use several baselines for performance comparison.

Graph-Enc-Dec (Avg). This baseline learns the graph encoder G_{Enc}^S and the sentence decoder G_{Dec}^S only on sentence corpus. It takes the average operation (as opposed to attention) over the three sets of features: $\mathcal{X}_o^k$, $\mathcal{X}_r^k$, and $\mathcal{X}_a^k$. During testing, we directly feed the image scene graph $\mathcal{G}^I$ to this model and get the image description.

Graph-Enc-Dec (Att*). This model shares the same setting with Graph-Enc-Dec (Avg) but replaces the average operation with a shared attention mechanism for all three sets (i.e., same attention for object, attribution, and relation).

Graph-Enc-Dec (Att). This model modifies the Graph-Enc-Dec (Att*) with an independent attention mechanism for each set of features.

Graph-Align. This is our final model. It is initialized with the trained parameters from Graph-Enc-Dec (Att) that uses separate attentions, and then it also learns the feature mapping functions using adversarial training.

7.3.3 Quantitative Results

Investigation on Sentence Decoding. In this experiment, we first train the network with Eq. (7.15), and then fine-tune it with Eq. (7.16) on the sentence corpus. Table 7.2 compares the results of three baseline models on the sentence corpus. It can be seen that the attention-based model performs better than the average-based model in all metrics, which demonstrates that weighting over features can better model the global dependency of features. Note that separate attention model for each set of features can significantly improve the performance. The inconsistent alignment of three kinds of features in Figure 7.4 also supports that we should treat these sets of features separately.

TABLE 7.2: Results for different sentence scene graph decoders on MSCOCO test split, where B@n refers to BLEU-n, M refers to METEOR, and C refers to CIDEr. All values are reported in percentage (bold numbers are the best results).

Methods	B@1	B@2	B@3	B@4	M	C
Graph-Enc-Dec(Avg)	84.3	71.8	58.8	47.1	31.0	129.4
Graph-Enc-Dec(Att*)	91.8	80.3	67.5	55.5	34.3	151.4
Graph-Enc-Dec(Att)	94.1	84.6	72.9	61.5	36.3	168.8

TABLE 7.3: Results for different baselines without GAN training on the test split of the MSCOCO.

Methods	B@1	B@2	B@3	B@4	M	C
Graph-Enc-Dec(Avg)	52.1	34.1	23.8	17.6	14.9	41.4
Graph-Enc-Dec(Att*)	54.3	37.0	26.8	20.3	15.9	47.2
Graph-Enc-Dec(Att)	56.0	33.6	20.1	11.9	17.0	48.5

Investigation on Unpaired Setting without GAN. Table 7.3 shows the comparisons among different baselines when no explicit cross-modal mapping of the

TABLE 7.4: Ablation studies of different GAN losses for Graph-Align model.

GAN Loss	Discriminator	B@1	B@2	B@3	B@4	C
BCE	$d_f \rightarrow d_f$	64.9	44.2	28.6	18.1	63.0
	$d_f \rightarrow 64$	66.0	46.0	30.3	19.7	65.5
	$d_f \rightarrow 1$	65.5	45.4	29.6	18.8	65.2
MSE	$d_f \rightarrow d_f$	65.3	44.8	28.9	18.3	62.9
	$d_f \rightarrow 64$	66.0	45.9	29.7	18.8	63.8
	$d_f \rightarrow 1$	58.4	36.3	21.7	12.6	46.7
GP	$d_f \rightarrow d_f$	66.1	46.1	30.3	19.5	65.5
	$d_f \rightarrow 64$	67.1	47.8	32.3	21.5	69.5
	$d_f \rightarrow 1$	64.5	44.2	28.5	17.9	61.1

features is done. By feeding the image scene graph directly to the trained scene graph encoder and the sentence decoder, we can achieve promising performance on the test set. Graph-Enc-Dec (Att) still achieves the best performance in all metrics. This is reasonable since both scene graphs and captions are high-level understandings of the image, and by capturing rich semantic information about objects and their relationships, scene graphs provide an effective way to connect an image to its natural language description. This finding also validates the feasibility of our approach to unpaired image captioning through the use of scene graphs. However, compared to the paired setup (see Table 7.6), these results are still inferior, meaning that only scene graph is not enough to achieve comparable performance.

Investigation on Unpaired Setting with GAN. To align the features from the image modality to the text modality, we use CycleGAN with our Graph-Enc-Dec(Att) model. Table 7.4 shows the comparisons of three kinds of GAN loss: binary cross-entropy (BCE) loss with logits (the vanilla GAN loss [71]), mean squared error (MSE) loss, and gradient penalty (GP) [152]. We also compare the results for using different output dimensions in the discriminator.¹

¹For example, for a dimension of 64, the output is a 64-dimensional vector, which is compared against an all-one vector of length 64 for a ‘Real’ input, and with an all-zero vector of length 64 for a ‘Fake’ input.

TABLE 7.5: The performances of using different feature mappings on MSCOCO test split. Shared GAN learns a shared feature mapping for three sets of features with CycleGAN. Single GAN concatenates the three kinds of embeddings together and learns a mapping with CycleGAN.

Methods	B@1	B@2	B@3	B@4	M	C
Shared GAN	60.7	41.3	26.9	17.6	20.0	60.1
Single GAN	61.8	42.1	27.3	17.7	20.1	61.2
Graph-Align	67.1	47.8	32.3	21.5	20.9	69.5

We can see that most of the CycleGAN variants improve the performance substantially compared to the results in Table 7.3. The GP with 64-dimension discriminator output achieves the best performance. Note that, when we set the output dimension to 1, the performance drops. This indicates that a strong discriminator is crucial for unpaired feature alignments. From the bottom row of Figure 7.4, we can see that with the help of the mapping module, the three kinds of embeddings are aligned very well, especially the attribute embedding (Figure 7.4f). It is also worth noting that the triplet features in Figure 7.4h are better aligned compared to the raw triplet features in Figure 7.4g.

To further demonstrate the effectiveness of the proposed three feature mapping functions, we conduct additional experiments in Table 7.5. It can be seen that treating the three set of embeddings ($\mathcal{X}_o^k$, $\mathcal{X}_r^k$, and $\mathcal{X}_a^k$) without distinction performs worse than Graph-Align.

Finally, Table 7.6 compares the results of the Graph-Align model with those of the existing unpaired image captioning methods [12, 80] on the MSCOCO test split. We can notice that our proposed Graph-Align achieves the best performance in all metrics. This demonstrates the effectiveness of our scene graph-based unpaired image captioning model.

TABLE 7.6: Performance comparisons on the test split of the MSCOCO dataset.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
<i>Paired Setting</i>								
SCST [153]	–	–	–	33.3	26.3	55.3	111.4	–
Stack-Cap [11]	78.6	62.5	47.9	36.1	27.4	56.9	120.4	20.9
SGAE (base) [154]	79.9	–	–	36.8	27.7	57.0	120.6	20.9
<i>Unpaired Setting</i>								
Language Pivoting [12]	46.2	24.0	11.2	5.4	13.2	–	17.7	
Adversarial+Reconstruction [80]	58.9	40.3	27.0	18.6	17.9	43.1	54.9	11.1
Graph-Align	67.1	47.8	32.3	21.5	20.9	47.2	69.5	15.0

FIGURE 7.4: Visualization of features in 2D space by t-SNE [4]. We plot the scatter diagrams for 1,500 samples.

FIGURE 7.5: Qualitative examples of different methods. In each example, the left image is the original input image; the middle is the image scene graph; the right image is the ground-truth sentence scene graph for comparison.

FIGURE 7.6: Examples of unpaired image captioning failure cases. Although the accuracy of image scene graph highly influences the performance of captioning results, our Graph-Align can still generate relevant image captions.

7.3.4 Qualitative Results

Figure 7.5 visualizes some examples of our models. We show the generated image descriptions using different models along with the ground-truth captions (bottom part). In the generated image and sentence scene graphs, we mark object, relation,

attribute nodes in orange, blue, and green, respectively. From these exemplary results, we observe that our method can generate reasonable image descriptions by aligning the unpaired visual-textual modalities with the help of scene graphs. Also, we observe that the number of attributes (words in green) in the sentence scene graph is less than that in the image scene graph. This observation potentially explains why there is a huge feature embedding gap between image and text in Figure 7.4e.

Figure 7.6 presents some failure cases of our Graph-Align model. We can see that the image scene graphs mainly focus on local regions/objects, while sentence scene graphs convey more information about the images. Such information misalignment leads to generating different captions.

7.4 Summary

In this chapter, we have proposed a novel framework to train an image captioning model in an unsupervised manner without using any paired image-sentence data. Our method uses scene graph as an intermediate representation of the image and the sentence, and maps the scene graphs in their feature space through cycle-consistent adversarial training. We used graph convolution and attention methods to encode the objects, their attributes, their relationships in a scene graph. Our experimental results based on quantitative and qualitative evaluations show the effectiveness of our method in generating meaningful captions, which also outperforms existing methods by a good margin.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

In this thesis, we have developed a few methods to bridge images and natural language with deep learning. The conclusions of this thesis are summarized as follows.

Firstly, Chapter 3 and Chapter 4 target at better sentence decoder for image captioning. To achieve this goal, a language-CNN based decoder and a coarse-to-fine decoder are proposed. More specifically, in Chapter 3, we model the hierarchical structure and long-term dependencies in word sequences with the CNN-based language model. In Chapter 4, we present a coarse-to-fine image captioning model which utilizes a stacked visual attention model in conjunction with multiple LSTM networks to generate complex and rich image descriptions. We perform extensive and rigorous experimental evaluations to validate the effectiveness of our proposed methods. Generally, the performance improvements of our methods are clearly observed when compared with other image captioning methods.

Secondly, we notice that conventional cross-modal feature embedding at the global semantic level is not sufficient to retrieve images with detailed local similarity (*e.g.*,

spatial layout) or sentences with word-level similarity. In Chapter 5, we incorporate the generative models (image-to-text and text-to-image) into the conventional cross-modal feature embedding, through which we can learn not only the high-level abstract representation and the local grounded representation of multi-modal data in a max-margin learning-to-rank framework. Our empirical results demonstrate that the combination of the grounded and the abstract representations can well match images and sentences with complex content.

Thirdly, in order to relax the requirement of paired image-caption data for automatic image caption generation, we address this problem with the help of a pivot language in Chapter 6. Our method couples an image-to-pivot captioning model with a pivot-to-target NMT model in a joint learning framework. In Chapter 7, we move a step further and adopt the scene graph as an explicit representation to bridge the gap between image and sentence domains. Unlike the pivot-based solution in Chapter 6 which still needs an image-pivot paired dataset, the scene graph-based framework in Chapter 7 is trained in an unsupervised manner without using any paired image-sentence data. Experimental results in these two chapters have shown that both of these two methods can generate meaningful captions.

In conclusion, five effective methods to bridge images and natural language have been proposed in this thesis. These solutions are also beneficial for other vision-language tasks, *i.e.*, video captioning and visual dialog, *etc.*

8.2 Future Works

Despite the progress made in this thesis, there are still many problems remaining. Here, we provide some recommendations for future research:

8.2.1 Supervised Image Captioning

In this thesis, the image captioning models in Chapter 4 and Chapter 5 are optimized by CIDEr score [11, 14, 155], which only calculates the similarity of the generated caption to the ground-truth captions. However, an image may contain multiple objects and multiple levels of detail. Thus, it is valuable to develop ways to generate multiple diverse captions describing different concepts and details in each image. Meanwhile, the image captioning models should also have the ability to generate human-like captions. Future research directions can go towards exploiting the inherent characteristics of human languages from other natural language datasets and generating more human-like captions.

The current image captioning models aim at generating only one sentence for one image. It is interesting to study how to generate a human-like paragraph for an image. Currently, some works have investigated this problem [156–158]. However, those image paragraphs are far from coherent, and they do not form a cohesive whole describing the entire image. Our proposed coarse-to-fine learning method in Chapter 4 could be applied to this problem in future work.

8.2.2 Cross-Modal Retrieval

Although our method in Chapter 5 has significantly improved the performance on visual-textual cross-modal retrieval, it is still a challenging problem due to the huge visual semantic discrepancy. The matching of images and sentences base on global abstract and local grounded features is still not sufficient for accurate cross-modal retrieval. Current higher-level visual understanding tasks such as scene graph construction have become popular research topics with enormous advancements. Scene graph provides a structured representation of objects and their pairwise relationships in the image [14]. Thus, a promising research direction is to use the scene graph as an explicit representation to match images and sentences.

On the other hand, using the scene graph structure to discover alignments between image and sentences still lacks global reasoning ability. Different image regions and semantic relationships would have different contributions for inferring the visual-textual similarity, and some of them are even redundant. For example, when a person describes what he/she see in the image using a natural language, it can be observed that the image descriptions not only contain objects but also involve their relationships and some high-level semantic concepts. Thus, the reasoning mechanism is important for current cross-modal retrieval systems.

8.2.3 Unpaired Image Captioning

The unpaired image captioning methods in Chapter 6 and Chapter 7 still need the sentence corpus close to the image captioning dataset [12, 14]. Learning an unpaired image captioning model on two completely independent datasets is a direction worth exploring. Meanwhile, instead of mapping the scene graphs in the image and the sentence feature space through cycle-consistent adversarial training, we can also explore other mapping methods such as optimal transport [159].

In Chapter 7, we align the images and the sentences via the scene graph. For text modality, most state-of-the-art statistical parsers are built for resource-rich languages [160–162]. However, one of the biggest drawbacks of statistical parsing is that data annotation is labor-intensive and time-consuming. Recently, some researchers find that the Transformer [163] can implicitly encode syntactic information such as dependency parse trees [164, 165]. When it comes to image, such approaches cannot be directly applied to visual relations since textual relations can often be captured by patterns, while visual relations are often local to an image. It is valuable to develop a method to generate visual relations without relying on any labeled data.

Author Publications

Journal Paper

- **Jiuxiang Gu***, Zhenhua Wang*, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent Advances in Convolutional Neural Networks. In *Pattern Recognition (PR)*, vol. 77, pp. 354-377, 2017.

Conference Papers

- **Jiuxiang Gu**, Joty Shafiq Rayhan, Jianfei Cai, Handong Zhao, and Gang Wang. Unpaired Image Captioning via Scene Graph Alignments. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- Xiangxi Shi, Jianfei Cai, Joty Shafiq Rayhan, and **Jiuxiang Gu**. Watch It Twice: Video Captioning with a Refocused Video Encoder. In *Proceedings of the 27th ACM International Conference on Multimedia (ACM Multimedia)*, 2019.
- **Jiuxiang Gu**, Handong Zhao, Zhe Lin, Sheng Li and Jianfei Cai, Mingyang Ling. Scene Graph Generation with External Knowledge and Image Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

- **Jiuxiang Gu**, Joty Shafiq Rayhan, Jianfei Cai, and Gang Wang. Unpaired Image Captioning by Language Pivoting. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.
- **Jiuxiang Gu**, Jianfei Cai, Joty Shafiq Rayhan, Li Niu, and Gang Wang. Look, Imagine and Match: Improving Textual-Visual Cross-Modal Retrieval with Generative Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- **Jiuxiang Gu**, Jianfei Cai, Gang Wang, and Tsuhan Chen. Stack-Captioning: Coarse-to-Fine Learning for Image Captioning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- **Jiuxiang Gu**, Gang Wang, Jianfei Cai, and Tsuhan Chen. An Empirical Study of Language CNN for Image Captioning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Bibliography

- [1] 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. [xvii](#), [10](#), [14](#), [15](#)
- [2] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. [xvii](#), [15](#)
- [3] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. [xvii](#), [15](#)
- [4] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, pages 2579–2605, 2008. [xix](#), [124](#)
- [5] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, pages 354–377, 2017. [1](#), [2](#), [24](#), [30](#), [32](#), [53](#), [105](#), [108](#)
- [6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. [1](#)

- [7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015. [1](#), [19](#), [88](#)
- [8] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015. [1](#)
- [9] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018. [1](#)
- [10] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen. An empirical study of language cnn for image captioning. In *ICCV*, 2017. [1](#), [62](#)
- [11] Jiuxiang Gu, Jianfei Cai, Gang Wang, and Tsuhan Chen. Stack-captioning: Coarse-to-fine learning for image captioning. In *AAAI*, 2017. [7](#), [18](#), [67](#), [73](#), [88](#), [105](#), [115](#), [123](#), [129](#)
- [12] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, and Gang Wang. Unpaired image captioning by language pivoting. In *ECCV*, 2018. [8](#), [23](#), [106](#), [118](#), [122](#), [123](#), [130](#)
- [13] Jiuxiang Gu, Jianfei Cai, Shafiq Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *CVPR*, 2018. [7](#), [105](#)
- [14] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, 2019. [1](#), [8](#), [23](#), [24](#), [107](#), [129](#), [130](#)
- [15] 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. [2](#), [27](#), [28](#), [38](#), [50](#), [76](#), [87](#), [117](#)

- [16] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. [25](#), [27](#), [28](#), [38](#), [89](#)
- [17] Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, et al. Ai challenger: A large-scale dataset for going deeper in image understanding. *arXiv preprint arXiv:1711.06475*, 2017. [2](#), [89](#), [96](#), [99](#), [100](#)
- [18] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016. [2](#), [5](#), [17](#), [19](#), [21](#), [27](#), [37](#), [41](#), [42](#), [43](#), [62](#)
- [19] 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. [41](#), [47](#), [105](#)
- [20] 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. [5](#), [19](#), [27](#), [41](#)
- [21] 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. [20](#), [41](#), [42](#), [43](#), [44](#), [47](#), [62](#)
- [22] Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. Multimodal pivots for image caption translation. In *ACL*, 2016.

- [23] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding long-short term memory for image caption generation. In *ICCV*, 2015. [17](#), [27](#), [39](#), [41](#)
- [24] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016. [2](#), [19](#), [20](#), [41](#)
- [25] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. [4](#), [27](#), [30](#), [41](#), [58](#), [67](#), [72](#), [76](#), [88](#), [97](#)
- [26] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. In *ACL*, 2015. [5](#), [20](#), [28](#)
- [27] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. [5](#), [20](#), [28](#)
- [28] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. [5](#), [21](#), [49](#), [55](#), [56](#), [57](#), [58](#), [61](#), [62](#), [73](#), [88](#), [100](#), [101](#), [115](#)
- [29] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018. [5](#), [23](#), [105](#)
- [30] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. [5](#), [22](#), [23](#), [107](#), [115](#)
- [31] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen. An empirical study of language cnn for image captioning. In *ICCV*, 2017. [7](#), [18](#), [21](#), [67](#), [88](#)
- [32] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, pages 215–243, 1968. [9](#)

- [33] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. 1982. [9](#)
- [34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, 1998. [10](#)
- [35] Robert Hecht-Nielsen. Theory of the backpropagation neural network. *Neural Networks*, 1(Supplement-1):445–448, 1988. [10](#)
- [36] Wei Zhang, Kazuyoshi Itoh, Jun Tanida, and Yoshiki Ichioka. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied optics*, 29(32):4790–4797, 1990. [10](#)
- [37] Xiao-Xiao Niu and Ching Y Suen. A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012. [10](#)
- [38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. [10](#), [12](#)
- [39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. [10](#), [12](#)
- [40] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. [10](#), [12](#)
- [41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [10](#), [15](#), [76](#), [98](#)

- [42] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade - Second Edition*, pages 9–48. 2012. [11](#), [33](#)
- [43] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. [11](#), [13](#)
- [44] Tao Wang, David J Wu, Andrew Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, 2012. [12](#)
- [45] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *ICML*, 2010. [12](#)
- [46] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. [12](#)
- [47] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2014. [12](#), [13](#)
- [48] Yichuan Tang. Deep learning using linear support vector machines. In *ICML Workshops*, 2013. [12](#)
- [49] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084–3104, 2012. [12](#)
- [50] Rob G. J. Wijnhoven and Peter H. N. de With. Fast training of object detection using stochastic gradient descent. In *ICPR*, 2010. [13](#)
- [51] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *NIPS*, 2010. [13](#)

- [52] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. [17](#), [34](#)
- [53] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [17](#)
- [54] 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. [17](#), [27](#), [41](#)
- [55] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *ACL*, 2014. [17](#)
- [56] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop*, 2014. [18](#)
- [57] Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutník, and Jürgen Schmidhuber. Recurrent highway networks. In *ICML*, 2017. [18](#)
- [58] 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. [19](#)
- [59] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 2015. [19](#), [48](#), [73](#)
- [60] 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 *ICML*, 2015. [20](#), [21](#), [41](#), [58](#), [62](#)
- [61] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *ICCV*, 2017. [20](#)
- [62] Chang Liu, Fuchun Sun, Changhu Wang, Feng Wang, and Alan Yuille. Mat: A multimodal attentive translator for image captioning. In *IJCAI*, 2017. [20](#), [21](#)
- [63] Yi Li, Kaiming He, Jian Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. [20](#)
- [64] Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016. [21](#), [47](#), [48](#), [56](#), [73](#)
- [65] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. [21](#), [36](#), [48](#), [73](#), [99](#), [118](#)
- [66] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. [21](#), [37](#), [39](#), [48](#), [73](#), [81](#), [99](#), [114](#), [118](#)
- [67] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *ICCV*, 2017. [21](#), [62](#)
- [68] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017. [21](#), [22](#)
- [69] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. In *ICLR*, 2017. [21](#)

-
- [70] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *ICCV*, 2017. [22](#)
 - [71] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. [22](#), [121](#)
 - [72] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. [22](#)
 - [73] Scott Reed, Zeynep Akata, Xincheng Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. [22](#), [75](#)
 - [74] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. [22](#), [75](#)
 - [75] 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. [22](#)
 - [76] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. [22](#)
 - [77] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017. [22](#)
 - [78] Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *EMNLP*, 2018. [23](#)

- [79] Amélie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, and Kevin Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. In *ICML*, 2018. [23](#)
- [80] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Unsupervised image captioning. In *CVPR*, 2019. [23](#), [106](#), [118](#), [122](#), [123](#)
- [81] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. [24](#)
- [82] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. [24](#)
- [83] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *NIPS Workshop*, 2014. [24](#), [67](#), [72](#)
- [84] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *ICLR*, 2016. [24](#), [67](#), [71](#), [72](#), [78](#), [84](#)
- [85] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. [24](#)
- [86] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. [24](#)
- [87] Satwik Kottur, Ramakrishna Vedantam, José MF Moura, and Devi Parikh. Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *CVPR*, 2016. [24](#)
- [88] Yuxin Peng and Jinwei Qi. Cm-gans: cross-modal generative adversarial networks for common representation learning. *TOMM*, page 22, 2019. [24](#)

- [89] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. In *BMVC*, 2018. [24](#), [81](#), [84](#), [85](#)
- [90] Andrej Karpathy, Armand Joulin, and Fei Fei F Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014. [25](#)
- [91] Xinyang Jiang, Fei Wu, Xi Li, Zhou Zhao, Weiming Lu, Siliang Tang, and Yueting Zhuang. Deep compositional cross-modal learning to rank via local-global alignment. In *ACMMM*, 2015. [25](#)
- [92] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *ICCV*, 2017. [25](#), [84](#)
- [93] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal lstm. In *CVPR*, 2016. [25](#), [84](#)
- [94] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. [27](#), [30](#)
- [95] 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 *CVPR*, 2011. [27](#), [88](#)
- [96] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *ACL*, 2014. [28](#), [32](#)
- [97] Mingxuan Wang, Zhengdong Lu, Hang Li, Wenbin Jiang, and Qun Liu. *gencnn*: A convolutional architecture for word sequence prediction. In *ACL*, 2015. [28](#)
- [98] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. [29](#), [38](#), [76](#)

- [99] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. [29](#)
- [100] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, 2016. [30](#)
- [101] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, 2017. [30](#)
- [102] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. Deep level sets for salient object detection. In *CVPR*, 2017. [30](#)
- [103] Jason Kuen, Zhenhua Wang, and Gang Wang. Recurrent attentional networks for saliency detection. In *CVPR*, 2016. [30](#)
- [104] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Multimodal neural language models. In *ICML*, 2014. [31](#)
- [105] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, 2014. [32](#)
- [106] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *NIPS*, 2015. [32](#)
- [107] Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutník, and Jürgen Schmidhuber. Recurrent highway networks. In *PMLR*, 2017. [34](#)
- [108] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *ACL*, 2014. [36](#), [99](#)
- [109] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. [37](#), [48](#), [111](#), [118](#)

-
- [110] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. [38](#), [59](#), [78](#), [98](#), [119](#)
 - [111] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. [38](#)
 - [112] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. [41](#), [43](#), [58](#), [62](#), [88](#), [97](#), [105](#)
 - [113] Zhilin Yang, Ye Yuan, Yuexin Wu, Ruslan Salakhutdinov, and William W Cohen. Encode, review, and decode: Reviewer module for caption generation. In *NIPS*, 2016. [41](#)
 - [114] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *NIPS*, 2016. [41](#), [62](#)
 - [115] Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. In *EMNLP*, 2016. [48](#)
 - [116] Jiatao Gu, Kyunghyun Cho, and Victor OK Li. Trainable greedy decoding for neural machine translation. In *EMNLP*, 2017. [48](#)
 - [117] Yuting Zhang, Kibok Lee, and Honglak Lee. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *ICML*, 2016. [49](#)
 - [118] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, 2017. [49](#)
 - [119] 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 vqa. In *CVPR*, 2018. [58](#)

- [120] 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. [62](#)
- [121] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *NIPS*, 2015. [67](#)
- [122] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, 2018. [107](#)
- [123] Bing Shuai, Henghui Ding, Ting Liu, Gang Wang, and Xudong Jiang. Toward achieving robust low-level and high-level scene parsing. *IEEE Transactions on Image Processing*, 28(3):1378–1390, 2018. [67](#)
- [124] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. [71](#)
- [125] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. [72](#)
- [126] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *ICLR*, 2017. [74](#)
- [127] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiao-gang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. [75](#)

-
- [128] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *CVPR*, 2015. [84](#)
- [129] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. [84](#)
- [130] 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. [88](#)
- [131] Hua Wu and Haifeng Wang. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, pages 165–181, 2007. [88](#)
- [132] Masao Utiyama and Hitoshi Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *NAACL*, 2007.
- [133] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017.
- [134] Yun Chen, Yang Liu, and Victor OK Li. Zero-resource neural machine translation with multi-agent communication game. In *AAAI*, 2018. [88](#)
- [135] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *SIGIR*, 2018. [99](#)
- [136] Daniel Cer, Christopher D Manning, and Daniel Jurafsky. The best lexical metric for phrase-based statistical mt system optimization. In *NAACL*, 2010. [101](#)

- [137] Xu Yang, Hanwang Zhang, and Jianfei Cai. Learning to collocate neural modules for image captioning. In *ICCV*, 2019. 105
- [138] English-speaking world, 2019. URL https://en.wikipedia.org/wiki/English-speaking_world. [Online; accessed 22-March-2019]. 105
- [139] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *ICLR*, 2018. 105
- [140] Handong Zhao, Quanfu Fan, Dan Gutfreund, and Yun Fu. Semantically guided visual question answering. In *WACV*, 2018. 107
- [141] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *ICCV*, 2019.
- [142] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *CVPR*, 2019.
- [143] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, 2017.
- [144] Xu Yang, Kaihua Tang, Hanwang Zhang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019. 107, 111, 118
- [145] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *CVPR*, 2017. 107
- [146] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 111
- [147] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 111, 118

-
- [148] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *ACL*, 2003. [111](#)
- [149] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *ACL*, 2015. [111](#)
- [150] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1): 32–73, 2017. [117](#)
- [151] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005. [118](#)
- [152] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017. [121](#)
- [153] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. [123](#)
- [154] Xu Yang, Hanwang Zhang, and Jianfei Cai. Shuffle-then-assemble: Learning object-agnostic visual relationship features. In *ECCV*, 2018. [123](#)
- [155] Xu Yang, Kaihua Tang, Hanwang Zhang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019. [129](#)
- [156] Moitreyia Chatterjee and Alexander G Schwing. Diverse and coherent paragraph generation from images. In *ECCV*, 2018. [129](#)

- [157] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*, 2017.
- [158] Xiaodan Liang, Zhiting Hu, Hao Zhang, Chuang Gan, and Eric P Xing. Recurrent topic-transition gan for visual paragraph generation. In *ICCV*, 2017. [129](#)
- [159] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE signal processing magazine*, 34(4):43–59, 2017. [130](#)
- [160] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *ACL*, 2005. [130](#)
- [161] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, 2014.
- [162] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. In *ACL*, 2015. [130](#)
- [163] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. [130](#)
- [164] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *NAACL*, 2019. [130](#)
- [165] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019. [130](#)