

*Doctor of Philosophy*  
*CRICOS Code: 058666A*  
*September 2019*

*Course Code:C02047*

# *Generating Descriptive and Accurate Image Captions with Neural Networks*

---

*Lingxiang Wu*

School of Electrical and Data Engineering  
Faculty of Engg. & IT  
University of Technology Sydney  
NSW2007, Australia



---

---

# Generating Descriptive and Accurate Image Captions with Neural Networks

---

---

*A thesis submitted in partial fulfilment of the requirements  
for the degree of*

Doctor of Philosophy

*by*

**Lingxiang Wu**

*to*

School of Electrical and Data Engineering  
Faculty of Engineering and Information Technology  
University of Technology Sydney  
NSW2007, Australia

September 2019



## CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Lingxiang Wu declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

**Signature:** Signature removed prior to publication.

**Date:** 2019/09/01



## ABSTRACT

Image captioning is to automatically describe an image with a sentence, which is a topic connecting computer vision and natural language processing. Research on image captioning has great impact to help visually impaired people understand their surroundings, and it has potential benefits for the sentence-level photo organization. Early work typically tackled this task by retrieval methods or template methods. Modern methods were mainly based on a combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). However, generating accurate and descriptive captions remains a challenging task. Accurate captions refer to sentences consistent with the visual content, and descriptive captions refer to those with diverse descriptions rather than plain common sentences. Generally, the vision model is required to encode the context comprehensively and the language model is required to express the visual representation into a readable sentence consistently. Additionally, the training strategy also affects the performance.

In this thesis, we develop methods and technics to generate descriptive and accurate image captions from three aspects with neural networks. First, we consider how to express the visual representation consistently in the language model. We propose a Recall Network that can selectively import the visual information using GridLSTM units in the RNN. This design efficiently prevents the RNN from deviating from the visual representation while gradually generating each word. Second, we explore a comprehensive visual representation in the vision model based on Graph Neural Networks (GCN). A grid-level visual graph is introduced to work collaboratively with a region-level graph, and GCN are applied to aggregate visual neighborhood information in the graphs. Finally, we design a new training strategy with Reinforcement Learning (RL) technics to boost captions' diversity. Unlike previous CNN-RNN frameworks, our framework contains an additive noise module which can manipulate the transition hidden states in the RNN. We train the noise module by our proposed noise-critic training algorithm.

Then we extend the caption generation into Modern Chinese Poetry creation from images. We identify three challenges in this task: (a) semantic inconsistency between

---

images and poems, (b) topic drift problems, and (c) frequent occurrence of certain words. Regarding the challenges, we develop a Constrained Topic-Aware Model. Particularly, we construct a visual semantic vector via image captions. A topic-aware generator is developed based on the Recall Network. An Anti-Frequency Decoding scheme is introduced to constrain the high-frequency characters.

Overall, we propose three novel modules for image captioning and an image-to-poetry framework. This thesis can facilitate the connection between computer vision and natural language, and it extends the generation into specific domains.



## ACKNOWLEDGMENTS

First and foremost, I must thank my supervisor Min Xu for her continuous support and guidance throughout my PhD. Min provided me much freedom on what I want to do, and thus made my PhD life pretty enjoyable. She is foresighted and professional, and she always provided helpful suggestions when I was confused. I am grateful that Min kindly supported me when I met difficulties in research or in personal life.

I am also thankful to Wenguan Wang and Shengsheng Qian for their generous guidance, encouragement and collaboration. It is a great opportunity to collaborate with them in two projects. I enjoyed very much discussing research with them, and I learned various aspects from them such like insightful thinking and technical writing.

There are many others I would like to express thanks during my two interns. Great thanks to Jinqiao Wang and Guibo Zhu who were my mentors in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. They introduced me in the research community and helped me a lot in my first research attempt. Also, a big thanks to Jianwei Cui, who is my mentor in Xiaomi Corporation. I was fortunate enough to work with Jianwei and had the freedom to choose my tendentious task.

My life in UTS would not be such enjoyable without my friends and colleagues in Sydney: Tianrong Rao, Haimin Zhang, Zhongqin Wang, Lei Sang, Ruiheng Zhang, Xiaoxu Li etc. Great thanks to Lei Sang for his generous ideas and professional collaboration. Thanks to Stuart Perry, Qiang Wu and Sean He for being my candidate assessment panels. Thanks for Dadong Wang for being my co-supervisor. Thanks to Jian Zhang for making their server available to me when we are short of GPUs at the beginning of my PhD. Thanks to Chandranath Adak (UTS) for providing this thesis template. Thanks to NVIDIA who provided a GPU for our research.

Finally, I owe many thanks to my boyfriend Xinyu Tang for his constant support and free help. He is an anonymous author in most of my papers. This thesis is dedicated to my parents and boyfriend for your love and support all these years.



## LIST OF PUBLICATIONS

### RELATED TO THE THESIS :

1. **Lingxiang Wu**, Min Xu, Jinqiao Wang, Stuart Perry, *Recall What You See Continually Using GridLSTM in Image Captioning[J]*, IEEE Transactions on Multimedia (TMM) 2019. (**Published**).
2. **Lingxiang Wu**, Min Xu, Shengsheng Qian, Jianwei Cui , *Image to Modern Chinese Poetry Creation via A Constrained Topic-Aware Model[J]*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). (**Under review**).
3. **Lingxiang Wu**, Min Xu, Wenguan Wang, Lei Sang, *Diverse Image Captioning with Trainable Noise and Dual Level Visual Graphs[J]*, IEEE Transactions on Image Processing (TIP). (**Under review**).

### OTHERS :

1. **Lingxiang Wu**, Jinqiao Wang, Guibo Zhu, Min Xu, Hanqing Lu, *Person re-identification via rich color-gradient feature[C]*, IEEE International Conference on Multimedia and Expo (ICME) 2016. (**Published**).
2. **Lingxiang Wu**, Min Xu, Guibo Zhu, Jinqiao Wang, Tianrong Rao, *Appearance features in Encoding Color Space for visual surveillance[J]*, Neurocomputing 308: 21-30 (2018). (**Published**).
3. Ruiheng Zhang, **Lingxiang Wu**, Yukun Yang, Wanneng Wu, Yueqiang Chen, Min Xu, *Multi-camera Multi-player Tracking with Deep Player Identification in Sports Video[J]*, Pattern Recognition. (**Under review**).



## TABLE OF CONTENTS

<b>List of Publications</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Main Challenges . . . . .	2
1.3 Contributions and Outlines . . . . .	3
<b>I Foundations</b>	<b>7</b>
<b>2 Related Works</b>	<b>9</b>
2.1 Traditional Captioning Methods . . . . .	10
2.1.1 Retrieval-based Methods . . . . .	10
2.1.2 Template-based Methods . . . . .	10
2.2 Modern Captioning Methods . . . . .	11
2.2.1 Neural Network Methods . . . . .	11
2.2.2 Dense Captioning and Styled Captioning . . . . .	12
2.2.3 Other Vision-language Tasks . . . . .	12
<b>3 Problem Statement</b>	<b>15</b>
3.0.4 Training with Cross Entropy Loss . . . . .	15
3.0.5 Training with Reinforcement Learning . . . . .	16

<b>II Proposed Methods for Image Captioning</b>	<b>17</b>
<b>4 Language Model: Recall Network</b>	<b>19</b>
4.1 Introduction . . . . .	20
4.2 LSTM Background . . . . .	21
4.3 Method . . . . .	22
4.3.1 Framework Overview . . . . .	22
4.3.2 Recall the Visual Information Continually in the Decoder . . . . .	24
4.3.3 Recall with the Depth Dimension LSTM . . . . .	25
4.3.4 Different Vision Models . . . . .	27
4.4 Experiments . . . . .	28
4.4.1 Datasets . . . . .	29
4.4.2 Evaluation Metrics . . . . .	30
4.4.3 Training Details . . . . .	30
4.4.4 Experiments on Full Images . . . . .	31
4.4.5 Experiments on Dense Captioning . . . . .	37
4.5 Summary . . . . .	38
<b>5 Vision Model: Dual GCN</b>	<b>41</b>
5.1 Introduction . . . . .	41
5.2 GCN Background . . . . .	42
5.3 Image captioning by two visual relation graphs . . . . .	43
5.3.1 Problem Formulation . . . . .	44
5.3.2 Region Visual Graph . . . . .	44
5.3.3 Grid Visual Graph . . . . .	45
5.3.4 GCN for Directed Labeled Graph . . . . .	45
5.3.5 RNN Generator . . . . .	46
5.4 Experiments . . . . .	46
5.4.1 Experimental Setup . . . . .	46
5.4.2 Comparative Models . . . . .	47
5.4.3 Ablative Analysis . . . . .	48
5.4.4 Comparison with State-of-the-art Models . . . . .	49
5.5 Summary . . . . .	50
<b>6 Training Strategy: Noise Agent</b>	<b>51</b>
6.1 Introduction . . . . .	51

6.2	RL Background . . . . .	53
6.3	Boosting Caption Diversity with a Noise Agent . . . . .	54
6.3.1	Gaussian Noise Agent . . . . .	54
6.3.2	Noise Critic Training . . . . .	55
6.3.3	Parallel Noise Decoding . . . . .	57
6.4	Experiments . . . . .	57
6.4.1	Experimental Setup . . . . .	57
6.4.2	Comparative Models . . . . .	57
6.4.3	Ablative Analysis . . . . .	58
6.4.4	Comparison with State-of-the-art Models . . . . .	59
6.4.5	Diversity Evaluation . . . . .	60
6.4.6	Qualitative analysis . . . . .	60
6.5	Summary . . . . .	61

### **III Extensions 63**

#### **7 Creating Modern Chinese Poetry from Images 65**

7.1	Introduction . . . . .	65
7.2	Background . . . . .	69
7.3	Creating Poetries From Images . . . . .	70
7.3.1	Problem Formulation . . . . .	71
7.3.2	Visual Semantic Vector Construction . . . . .	71
7.3.3	Topic-Aware Poetry Generation . . . . .	74
7.3.4	Anti-frequency Decoding . . . . .	77
7.4	Experiments . . . . .	79
7.4.1	Dataset . . . . .	79
7.4.2	Implementation Details . . . . .	79
7.4.3	Evaluation Metrics . . . . .	80
7.4.4	Comparative Methods . . . . .	81
7.4.5	Quantitative Evaluation . . . . .	82
7.4.6	Qualitative Evaluation . . . . .	86
7.4.7	Effects of Hyperparameter $\lambda$ . . . . .	87
7.5	Summary . . . . .	88

#### **8 Conclusion 89**

## TABLE OF CONTENTS

---

8.1	Summary . . . . .	89
8.2	Future Work . . . . .	90
	<b>Reference</b>	<b>93</b>



## LIST OF FIGURES

FIGURE	Page
1.1 Examples of object recognition, full image captioning and dense captioning. . . . .	2
1.2 Suboptimal descriptions from automatic generators. . . . .	3
3.1 The basic CNN-RNN framework for image captioning. . . . .	15
4.1 An illustration of the proposed model. Our model contains two LSTMs at each step. We use the temporal LSTM to transmit the sequential caption states and the depth LSTM to integrate the visual information. . . . .	23
4.2 Three different decoder units. (a): The image representation is only processed once with the LSTM. (b): The image representation and word embedding are concatenated as extra inputs for the LSTM. (c): Our Recall Network. . . . .	26
4.3 Loss comparison on validation split along the training process. . . . .	32
4.4 CIDEr score comparison on validation split along the training process. . . . .	33
4.5 Captions generated by the Recall Network. The comparison between the Recall Network and the conventional encoder-decoder model shows that our model generates more accurate and detailed captions. The comparison between the generated results and the human annotated ground truth shows that further work should be carried out for more comprehensive content captions. . . . .	36
4.6 Examples generated by the Recall Network in dense captioning. Each caption corresponds to the bounding box in the same color. . . . .	38
5.1 Region-level and grid-level visual graph construction. . . . .	42
5.2 The proposed CNN-GCN-RNN model. The image is represented by a region level graph and a grid level graph. GCN is utilized to encode neighbourhood information. . . . .	43
6.1 (a) Existing RL framework in captioning. (b) Our proposed one with the noise module. . . . .	52

6.2	The transition details in the noise-added model. . . . .	54
6.3	Generation examples with naive noise and adaptive noise. . . . .	56
6.4	CIDEr and BLEU-4 log on validation set while training the noise agent . . .	59
6.5	Generated captions from ResNet baseline, our G-noise model and the Ground Truth (GT). . . . .	61
7.1	Three challenges in the image-poetry generation. (a) Semantic inconsistency between the image and the poetry. (b) Topic drift in the subsequent lines. (Relative words are shown in bold.) (c) Frequent occurrence of certain words (shown in bold). . . . .	67
7.2	The framework of our proposed CTAM . . . . .	70
7.3	Comparison on keywords obtained from two different visual models. . . . .	74
7.4	The topic-aware LSTM details. . . . .	75
7.5	Comparison results. (a) The occurrences of seven frequently appearing characters generated by the model w/ and w/o anti-frequency decoding. (b) The percentage of generated poetries containing keywords. . . . .	84
7.6	The number of poetry keywords and caption words. . . . .	85
7.7	Visualization of image captions and related keywords. . . . .	87
7.8	Visualization of generated results of Sequence model and the CTAM. We manually highlight relative words in poems, and present their generation probabilities in blue bars. Human evaluation results on Semantic consistency (S), Readability (R), Poeticness/Aesthetics (P) and the Average score (A) are shown in orange stripes. . . . .	87
7.9	Frequency of character "I" with respect to variant hyperparameter $\lambda$ . . . . .	88
8.1	An example with detailed entities. This figure is from RichCaptioning[78] . .	91

## LIST OF TABLES

TABLE	Page
4.1 Dataset size summary . . . . .	29
4.2 Comparison with the baseline methods . . . . .	32
4.3 Comparison with the Stacked LSTM . . . . .	33
4.4 Comparison with the state-of-the-art methods on MSCOCO dataset for Offline Evaluation. . . . .	34
4.5 Online Evaluation Performance on the MSCOCO Test Server . . . . .	35
4.6 Dense Captioning Performance on the Visual Genome Dataset . . . . .	38
5.1 GCN Model Performance comparisons on MSCOCO offline set . . . . .	48
5.2 Comparisons on the updating times of the Grid GCN. . . . .	49
5.3 Performance comparisons with the state-of-the-art methods on MSCOCO offline set . . . . .	49
5.4 Performance evaluation on MSCOCO online Test Server . . . . .	50
6.1 Performance comparisons for ablation study on MSCOCO offline set . . . . .	58
6.2 Performance comparisons with the state-of-the-art methods on MSCOCO offline set . . . . .	59
6.3 Performance evaluation on MSCOCO online Test Server . . . . .	60
6.4 Diversity Evaluation . . . . .	60
7.1 Listing of notations in this chapter . . . . .	71
7.2 Human Evaluation Results . . . . .	82
7.3 Ablation study with BLEU-1 (%) . . . . .	84
7.4 Image captioning performance . . . . .	86



## INTRODUCTION

## 1.1 Overview

One of the ultimate goal in Artificial Intelligence is to enable computers to understand the real visual world and endow them the abilities to serve or communicate with humans. The study of computer vision aims to represent and understand the complex visual world. Generally, the communication between humans is through the natural language. The study of natural language processing is meaningful for human-computer interaction, and it is also important to achieve the “ultimate” goal. Interactions between images and texts is a growing research field. Early work includes generating single words or fixed phrases from images [6, 18]. Image captioning refers to generating a natural sentence description given an image, which is a core topic to achieve the “ultimate” goal. Automatic captioning involves both the visual context understanding in computer vision and the sentence generation in natural language processing. An example of object recognition, full image captioning and dense captioning (to detect salient regions and generate regional descriptions) is presented in Fig. 1.1. This thesis seeks methods to generate accurate and descriptive image captions.

The research on image captioning has great significance on multimedia/scene understanding, and it has potential applications that can benefit the social communities. For example, it can help the visually impaired people understand their surroundings. In conjunction with speech synthesis system, the intelligent research can turn the visual world into an audible experience. Another potential application can be the sentence-level



Figure 1.1: Examples of object recognition, full image captioning and dense captioning.

photo organization. It would be better if the photo platform, e.g., Instagram, Google Photos, can organize our photo with semantic stories rather than the time stamp only. Also, these platform can provide photo diaries for customers. Image captioning provide a fundamental strategy for these applications.

Recent advances have revealed that a bunch of effective caption generators are based on the encoder-decoder framework. These neural network methods [32, 81, 95] are inspired by recent advances in machine translation. In this framework, an encoder network, a Convolutional Neural Network (CNN), encodes an image into a context vector. Then, the decoder network, a Recurrent Neural Network (RNN), decodes the context vector into a sequence of words. Based on this CNN-RNN framework, a bunch of variants embed the attention mechanism and reinforcement learning technics. In this thesis, we develop innovative modules based on the CNN-RNN framework.

## 1.2 Main Challenges

Humans find it easy to understand the visual scenes and express it in natural language. For instance, a single glance is enough for humans to reason the relationship between the objects and describe it in detail. While, it is a challenging task for the computer system to generate accurate and descriptive captions. Accurate captions refer to sentences consistent with the visual content, and descriptive captions refers to those with diverse descriptions rather than plain common sentences.

This task requires the model to encode detailed understanding of the image contents including major objects and relationship among objects. A suboptimal example is shown in Fig 1.2(a) where the model fails to describe the relationship between the “man” and the “motorcycle” correctly. Besides, it requires appropriate expression in the natural

language sentence. The suboptimal example in Fig. 1.2(b) reads not fluent enough due to the phrase “clutter and clutter”. Additionally, this task is challenging due to the lack of diversity and details. Going through a bunch of captions from a generator, you can find that most of the “bathroom” is with “a toilet and a shower” as shown in Fig. 1.2(c)

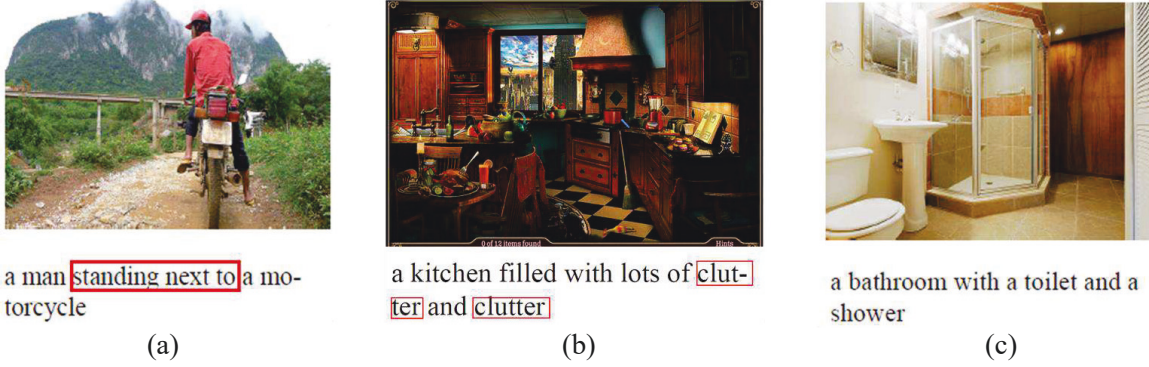


Figure 1.2: Suboptimal descriptions from automatic generators.

## 1.3 Contributions and Outlines

In thesis, we aim to develop novel models to generate accurate and descriptive captions for images. To achieve this goal, the **vision model** is required to encode the context comprehensively, and the **language model** is required to express the visual representation into a sentence consistently. Additionally, the **training strategy** also affects the generation performance. Thus, we develop methods and technics to generate image captions from these three aspects with neural networks. At last, we extend the image captioning to **image-to-poetry generation**.

In Chapter 2, we provide the literature review of image captioning.

In Chapter 3, based on the CNN-RNN framework, we formulate our task in two scenarios: training with cross entropy loss and training with reinforcement learning technics.

In **Chapter 4**, we consider how to express the visual representations consistently in the language model. Although the conventional CNN-RNN model [81] and its variants provide a practical way for the vision-text transformation, it still has some limitations on image-consistent captions. Existing methods use the image features in three ways: 1) they inject the encoded image features into the decoder only once at the initial step,



which does not enable the rich image content to be explored sufficiently while gradually generating a text caption; 2) they concatenate the encoded image features with text as extra inputs at every step, which introduces unnecessary noise; and 3) they use the attention mechanism [95], which increases the computational complexity due to the introduction of extra neural nets to identify the attention regions. To prevent the decoder from deviating from the original image content, we propose a Recall Network. Specifically, at each step, there are a temporal LSTM and a depth LSTM in the language model. We input the visual information as the latent memory along the depth dimension LSTM, and thus the decoder is able to admit the visual features dynamically through the inherent LSTM structure. The content of this chapter is based on the work published in TMM<sup>1</sup>

In **Chapter 5**, we explore a comprehensive visual representation in the vision model. Existing generation process is easy to omit rich visual information in the image. In this chapter, visual graphs and Graph Neural Networks (GCN) are applied to explore the visual context. However, region-based visual graphs fail to explore the rich background visual context because detection algorithms tend to only consider salient areas in an image. We introduce a grid-level visual graph to work collaboratively with a region-level graph, which can encode fine-grained background context ignored by regional objects. The grid level graph is constructed with designed labels on each eight-connected region of the full image’s feature map. Then, GCN are applied to aggregate visual neighborhood information in the graphs. The content of this chapter is based on the work submitted to TIP<sup>2</sup>.

In **Chapter 6**, we explore a novel training strategy. To pursue accuracy and diversity in captions, we introduce an additive noise module in the RNN decoder, which can manipulate the transition hidden states. In this way, we add adaptive perturbation in the word distribution. To train such a module, we regard the noise module as an agent with a stochastic gaussian policy, and generating noise is regarded as the action. The policy net is optimized by an introduced noise-critic training algorithm, where we use noiseless results as the baselines in the REINFORCE algorithm. The content of this chapter is based on the work submitted to TIP<sup>2</sup>.

In **Chapter 7**, we extent the image captioning to modern Chinese generation from images. We figure out three major challenges in this task: semantic inconsistency, topic drift and word re-appearance. In regard to these challenges, we develop a Constrained

---

<sup>1</sup>Lingxiang Wu, Min Xu, Jinqiao Wang, Stuart Perry, *Recall What You See Continually Using GridLSTM in Image Captioning*, IEEE Transactions on Multimedia (TMM).

<sup>2</sup>Lingxiang Wu, Min Xu, Wenguan Wang, Lei Sang, *Diverse Image Captioning with Trainable Noise and Dual Level Visual Graphs*, IEEE Transactions on Image Processing.



Topic-Aware Model. To achieve the semantic consistency between images and poems, we construct visual semantic vectors via the image captions. To handle the topic drift problem, the Recall Network introduced in Chapter 4 is extended to a topic-aware generator. Then, an Anti-Frequency Decoding scheme that is based on Mutual Information (MI) is introduced to constrain high-frequency characters. The content in this chapter is based on our work submitted to ACM TOMM<sup>3</sup>.

---

<sup>3</sup>Lingxiang Wu, Min Xu, Shengsheng Qian, Jianwei Cui, *Image to Modern Chinese Poetry Creation via A Constrained Topic-Aware Model.*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM).



# **Part I**

## **Foundations**



## RELATED WORKS

Existing captioning methods can be summarized into traditional methods and modern captioning methods. Early work focus on the traditional ones, which include retrieval-based methods and template-based methods. The retrieval-based methods regard image captioning as a retrieval task. Given a query image, the retrieval-based methods find the most relative sentence by a designed similarity metric. The disadvantage of this kind of methods is that they are unable to describe new combinations of objects or novel scenes because they only feed back predetermined sentences. The template-based methods generally divide the image caption task into sub problems, including phrase extraction, phrase retrieval and composing phrases into captions. However, the template is generally suboptimal due to the lack of syntactic variability.

Modern captioning methods mostly generate captions with neural networks. Basically, convolutional neural networks encode the image context into a representation and recurrent neural networks decode it into a sequence of words. After that, some methods tried to integrate various attention mechanisms into the vanilla CNN-RNN framework. Afterwards, some methods tended to tackle captioning task with Reinforcement Learning (RL) since the RL technics can optimize the non-differentiable metrics. Another trends focus on dense captioning as well as styled captions. In dense captioning, the models are required to both detect salient regions and generate descriptions for each region. Besides, some vision-language tasks such as storytelling, visual grounding etc. also attract the research attentions.

This chapter provides literature review for traditional and modern captioning meth-

ods in detail.

## 2.1 Traditional Captioning Methods

### 2.1.1 Retrieval-based Methods

This sort of methods [21, 30, 62] regard image captioning as a retrieval task. These methods map images and text into a common vector space, or define a similarity metric score for ranking. Given a query image, captions are retrieved based on this ranking.

In [62], a web-scale captioned dataset was collected. Given a query image, candidate match images were retrieved from this dataset using global image descriptors. Then, high level information related to image content, *e.g.* objects, scenes, etc, was extracted. Images in the match set were re-ranked based on image content. At last, the best fitting caption was returned for the query. The quality of captions is seriously affected by the dataset size. In [21], similarity between a sentence and an image was evaluated directly. All images and text were represented as linguistically-motivated semantic triplets <object, action, scene>, and similarities were computed in a meaning space. Two mapping, the mapping from the image space to the meaning space and the mapping from the sentence space to the meaning space, were learned through structure learning. However, the triplet is suboptimal in that their sentence model is oversimplified and lack syntactic variation. In [30], Hodosh *et al.* introduced a new benchmark collection consisting of 8,000 images that are each paired with five different captions. Kernel Canonical Correlation Analysis was adopted to associate images and captions.

The deficiency for retrieval-based methods is that they are insufficient to describe new combinations of objects or novel scenes because it only feeds back existing sentences. This deficiency has motivated a large number of generative approaches introduced as follows.

### 2.1.2 Template-based Methods

This sort of methods generate sentences by a template or a grammar model with linguistic constraints. Images are generally explained by detected objects, adjectives and spatial relationships (prepositions).

In [44], image descriptions were generated using web-scale n-grams given computer vision input. In particular, given an input image, vision system extracted objects, attributes and spatial relationships within the image by object detectors, attribute clas-

sifiers and hand designed preposition functions respectively. Then, they found n-gram phrases in designed pattern from the Google Web 1T data. At last, a new phrase was composed using dynamic programming. In [41], Conditional Random Field (CRF) was utilized to correspond objects, attributes and propositions within an image. Then a sentence-related label could be predicted. In [79], captions were generated through three steps: phrase extraction from image descriptions, learning phrases using images and extracted phrases, and caption generation from estimated phrases.

Template-based methods generally handle image captioning problem by dealing with sub-problems. Moreover, these methods rely on the fixed template, but the template is suboptimal due to the lack of syntactic variability.

## 2.2 Modern Captioning Methods

### 2.2.1 Neural Network Methods

Recent research efforts focus on the neural-network-based methods which were mainly based on a CNN-RNN framework. These neural-network-based methods were inspired by recent advances in machine translation. In [81], Vinyals *et al.* proposed to encode the high level visual features with Convolutional Neural Networks (CNN) and decode the CNN representation into a word sequence with Recurrent Neural Networks (RNN). Normally, the CNN are pre-trained on the ImageNet dataset [69], and the RNN constructed with LSTM receive the image information through the initial interactions and predict subsequent words given the previous word. Generally, components such as LSTM [29] or GRU [12] can be used in the RNN language models. Based on this framework, in [102], image attributes from a separate predictor were also fed into the RNN language model to provide semantic context.

On the basis of the CNN-RNN framework, a bunch of methods [2, 95, 99] tended to integrate attention mechanisms into the the vanilla CNN-RNN framework. In [95], Xu *et al.* proposed soft attention and hard attention. In [99], a number of review steps were preformed between the encoder and the attentive decoder. Bottom-up and top-down attention were integrated for image captioning as well as visual question answering in [2]. In [22], image representation with localized regions at multiple scales was encoded through the encoder, and the decoder can generate words sequentially as well as focus on these regions.

In addition to the end-to-end training frameworks introduced above, there are some

multiple-stage methods. In [20], Fang *et al.* used multiple instance learning to train visual detectors for words that commonly occur in captions, and then developed a model to generate sentences with these words through maximum-entropy training. In [3], Hendricks *et al.* focused on compositional models trained on external datasets to deal with the zero-shot problem (when there are no image-sentence paired samples for training). They first trained a lexical classifier and a language model independently on the unpaired image data and unpaired text data, and then they combined them jointly into a deep caption model that is trained on an image-sentence dataset.

Afterwards, some method tended to tackle captioning task with Reinforcement Learning. In [68], Rennie *et al.* tried to optimize the network with policy gradient and proposed an efficient self-critical baseline for training. In [67], a policy network and a value network were trained collaboratively to generate captions.

### 2.2.2 Dense Captioning and Styled Captioning

In [35], a novel task was proposed, dense image captioning. The model is required not only to detect the localization of salient objects, but also to generate a caption for each region detected. Johnson *et al.* developed a model on the basis of Faster R-CNN [66] to learn region proposals and region features jointly, then they fed these features into an RNN language model. This model enabled multiple region descriptions in one image, and makes captions more specific.

Different from generating factual captions, some works tend to generate styled descriptions for images. In StyleNet [23], Gan *et al.* proposed to generate humorous or romantic image descriptions using a standard image-caption dataset and an external monolingual text dataset. The humorous/romantic descriptions were generated by updating certain LSTM parameters while training on the monolingual corpus. In SentiCap [54], Mathews *et al.* proposed to generate captions with positive or negative sentiments. This was achieved by two labeled datasets: a standard image-caption dataset and a dataset containing captions with sentiments.

### 2.2.3 Other Vision-language Tasks

In addition to the above mentioned captioning tasks, there are some other vision-language tasks attracting research attentions, e.g., visual storytelling, visual grounding, visual dialog and visual paragraph generation.



In visual storytelling, a narrative paragraph is generated given a photo stream. The challenge is that the sequence of images generally depict events as they occur or change. In [86], adversarial training was utilised to model the ordered image sequence and generate the relevant descriptions. Different from visual storytelling, visual paragraph generation focus on producing a coherent paragraph to describe the visual content of one image. The challenge lays on how to encapsulate multiple gists/topics in an image, and then describe the image from one topic to another with a coherent structure. In [87], Convolutional Auto-Encoding was employed for topic modeling on the regional features of an image. In visual dialog [105], an image is given as context input, associated with a caption and dialog history. The goal is to answer questions posed in natural language about images, or recover a follow-up question based on the dialog history. Additionally, visual grounding [36] tends to locate the components of a structured description in an image.

In this thesis, we focus on generating factual captions with the neural network based method.



## PROBLEM STATEMENT

Our models are based on the CNN-RNN framework. In this chapter, we first introduce the basic framework trained with cross entropy loss. Then, we introduce how to formula the captioning problem in the Reinforcement Learning.

### 3.0.4 Training with Cross Entropy Loss

Generally, a captioning generator receives an image  $\mathbf{I}$  as the input and outputs a sentence  $\mathbf{S}$  to describe the visual content.  $\mathbf{S}$  can be represented as a sequence of words,  $\mathbf{S} = \{\mathbf{w}_1, \mathbf{w}_2 \cdots \mathbf{w}_N\}$ , where  $\mathbf{w}_t$  denotes the  $t$ -th word. In the basic CNN-RNN framework, the image  $\mathbf{I}$  is normally encoded into a vector or matrix  $\mathbf{v}_{CNN}$  via the Convolutional Neural Networks. Then, the  $\mathbf{v}_{CNN}$  is injected into the RNN through the initial step and the RNN decoder generates a caption word by word. Normally, the RNN consists of a sequence of LSTM units. At each step, with the LSTM output, we can obtain the probabilities over the words in the dictionary. An illustration is shown in Fig. 3.1

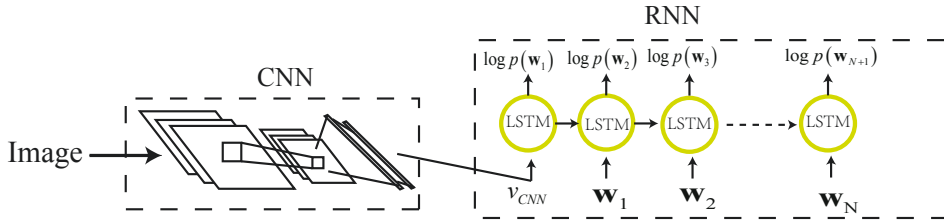


Figure 3.1: The basic CNN-RNN framework for image captioning.

During the training,  $(\mathbf{I}, \mathbf{S})$  is given as a training example pair. The image caption model (parameterized by  $\theta$ ) can be optimized by minimizing the cross entropy loss:

$$(3.1) \quad L(\theta) = -\sum_{t=1}^N \log p(\mathbf{w}_t | \mathbf{I}, \mathbf{w}_0, \dots, \mathbf{w}_{t-1}; \theta).$$

### 3.0.5 Training with Reinforcement Learning

The generator is trained with cross entropy loss, while, it is generally evaluated with non-differentiable metrics such as CIDEr, BLUE at testing time. Recently it has been shown that policy-gradient methods in reinforcement learning can be utilized to train end-to-end systems directly with the non-differentiable metrics.

In the RL setting, the generator can be viewed as an agent that interacts with an external environment (words and image features). The parameters in the CNN-RNN generator define a stochastic policy, and the action is to predict the next word given the probability. After executing an action, the agent receives a scalar reward. The training goal is to minimize the negative expected rewards (e.g., the NLP evaluation metrics):

$$(3.2) \quad L(\theta) = -\mathbb{E}_{S \sim \pi_\theta} [\sum_t r_{w_t}].$$

To approximate the policy, some deep reinforcement technics are utilized. For example, with REINFORCE algorithm [91], the gradient can be computed as:

$$(3.3) \quad \Delta_\theta L(\theta) \approx -(r_S - b) \Delta_\theta \log p_\theta(S).$$

$b$  represents a baseline which can reduce variance. In SCST [68], an efficient baseline is proposed. This self-critic algorithm[68] uses the output of greedy sample generation as the baseline. Thus, sampled outputs outperforming the greedy sampled outputs can get the positive return. Normally, the model trained with the self-critic algorithm can outperform the one trained with cross entropy loss.

# **Part II**

## **Proposed Methods for Image Captioning**



## LANGUAGE MODEL: RECALL NETWORK

In this chapter, our goal is to develop a language model which can make the caption consistent with the image content. The existing encoder-decoder model and its variants, which are the most popular models for image captioning, use the image features in three ways: 1) they inject the encoded image features into the decoder only once at the initial step, which does not enable the rich image content to be explored sufficiently while gradually generating a text caption; 2) they concatenate the encoded image features with text as extra inputs at every step, which introduces unnecessary noise; and 3) they using an attention mechanism, which increases the computational complexity due to the introduction of extra neural nets to identify the attention regions. Different from the existing methods, in this chapter, we propose a novel network, Recall Network, for generating captions that are consistent with the images. The Recall Network selectively involves the visual features by using a GridLSTM and thus is able to recall image contents while generating each word. By importing the visual information as the latent memory along the depth dimension LSTM, the decoder is able to admit the visual features dynamically through the inherent LSTM structure without adding any extra neural nets or parameters. The Recall Network efficiently prevents the decoder from deviating from the original image content. To verify the efficiency of our model, we conducted exhaustive experiments on full and dense image captioning. The experiment results clearly demonstrate that our Recall Network outperforms the conventional encoder-decoder model by a large margin and that it performs comparably to the state-of-the-art methods.

## 4.1 Introduction

Recent advances have revealed that encoder-decoder frameworks [32, 81, 95] can achieve end-to-end training and are capable of conveying the image details in one sentence. These neural-network-based methods are inspired by recent advances in machine translation. In these frameworks, an encoder network, a Convolutional Neural Network (CNN), encodes an image into a context vector. Then, the decoder network, a Recurrent Neural Network (RNN), decodes the context vector into a sequence of words. Generally, components such as LSTM [29] or GRU [12] can be used in the RNN language models.

Although the conventional encoder-decoder model [81] provides a practical way to construct the visual and text data flow, it still has some limitations on image-consistent captions. The image features in this model are injected only once into the decoder at the initial step, which means that the decoder may deviate from the image contents while gradually generating the text caption. Thus, the rich visual details cannot be sufficiently taken into account. Although, in [32], the image features are concatenated with word embedding and are fed into the decoder as extra inputs at each time step, the performance is not satisfying, mostly due to the unnecessary noise involved [82]. Most recently, the attention encoder-decoder models [95] [99] have been introduced to solve the bottleneck in [81]. However, the attention mechanism has to rely on extra neural nets to emphasize the attention regions. Ideally, the visual information should be considered continually and properly for generating each word.

In this chapter, to deal with the deficiencies in the existing encoder-decoder frameworks, we propose a novel network, Recall Network, for generating image-consistent captions. The main goal of the Recall Network is to recall the visual information continually and properly when generating each word. We are inspired by GridLSTM [37] which has LSTM cells in multiple dimensions. In [37], it has been found that having LSTM units along the depth dimension is more effective than not having them in some natural language processing tasks. To achieve our goal, we construct the network with the GridLSTM and modify the GridLSTM to adaptively take image features into account through memory cells of the depth LSTMs. In this network, the image features are initially input as an overview of the whole image content and then recalled continually at each step as the “previous memory” along the depth dimension in the GridLSTM. As opposed to roughly admitting the still visual information, we make use of the memory cell in the depth LSTM structure to selectively forget and update the visual information according to the corresponding word. The design yields a simple yet efficient network



that adaptively involves the visual information at each step without increasing any extra nets or trainable parameters.

The main contributions of this chapter can be summarized as follows:

- To prevent the decoder from deviating from the visual guidance, we introduce an innovative encoder-decoder model that recalls the image information continually in the decoder to generate image-consistent captions.
- We modify the GridLSTM to adaptively take image features into account through the depth LSTM memory cell, which achieves effective functionality without extra nets or parameters being introduced.
- Exhaustive experiments are conducted on both full image captioning and dense captioning. Our Recall Network outperforms the conventional encoder-decoder model by a large margin and performs comparably to the-state-of-art methods.

## 4.2 LSTM Background

In this section, we briefly introduce the background of LSTM. A Recurrent Neural Network is a neural network that processes a sequence of entities, while a Long Short-Term Memory (LSTM) performs as an activation function unit in the RNN. The conventional RNN processes the input sequence  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$  and output sequence  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$  through an internal hidden state  $\mathbf{h}$  as follows:

$$(4.1) \quad \mathbf{h}_t = g_\theta(\mathbf{h}_{t-1}, \mathbf{x}_t),$$

$$(4.2) \quad \log p(\hat{\mathbf{y}}_t | \mathbf{x}_{<t}) = f_\theta(\mathbf{h}_t),$$

where  $g_\theta$  is a nonlinear activation function parameterized by a set of parameters  $\theta$ .  $p(\hat{\mathbf{y}}_t | \mathbf{x}_{<t})$  is the probability of prediction  $\hat{\mathbf{y}}_t$  at time  $t$ .  $f_\theta$  can be a parametric function learned jointly in the whole framework.

The conventional RNN encounters a problem of vanishing gradient, where the backward signals may decay sharply through the chain. A more sophisticated activation function, i.e. the LSTM, is introduced to avoid this problem through gating and memory cells. The LSTM updates and outputs as follows:

$$\begin{aligned}
 \mathbf{i}_t &= \sigma(\mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{W}_i \mathbf{x}_t + \mathbf{b}_i), \\
 \mathbf{f}_t &= \sigma(\mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{W}_f \mathbf{x}_t + \mathbf{b}_f), \\
 \mathbf{o}_t &= \sigma(\mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{W}_o \mathbf{x}_t + \mathbf{b}_o), \\
 \tilde{\mathbf{C}}_t &= \tanh(\mathbf{U}_C \mathbf{h}_{t-1} + \mathbf{W}_C \mathbf{x}_t + \mathbf{b}_C), \\
 \mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t, \\
 \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{C}_t),
 \end{aligned}
 \tag{4.3}$$

where  $\mathbf{i}_t$ ,  $\mathbf{f}_t$ , and  $\mathbf{o}_t$  are the input gate, forget gate, and output gate, respectively.  $\mathbf{C}_t$  is the memory cell.  $\mathbf{W}$ ,  $\mathbf{U}$  and  $\mathbf{b}$  are parameter metrics to be learned. The LSTM mechanism has two important properties. The memory cell is obtained by a linear transformation of the previous memory cell and gates, rather than a nonlinear function, which ensures that the signal in backward propagation does not decay sharply. The memory cell also involves new information to update, and drop information that is selected to be forgotten, which acts as an attention system [37].

The differences between the LSTM and the modified GridLSTM unit in image captioning will be presented in detail in Sec. 4.3.3.

## 4.3 Method

We start by briefly presenting our model’s overview and the conventional encoder-decoder framework for image captioning in Sec. 4.3.1 and then introduce the details of our Recall Network in Sec. 4.3.2 and Sec. 4.3.3. As we conduct the experiments on both full image captioning and dense captioning, which are slightly different in terms of the visual model, we briefly review the different visual models in Sec. 4.3.4.

### 4.3.1 Framework Overview

An overview of our approach is depicted in Fig. 4.1. Our model is based on the conventional encoder-decoder model. Rather than only using temporal LSTM units to model the probability of the target words, we further exploit the depth LSTM to interact with image views.

In machine translation, a sentence in a source language can be encoded into a context vector. Then, the decoder network translates it into a sentence in the target language. In the conventional encoder-decoder framework for image captioning, given an image  $\mathbf{I}$ , the

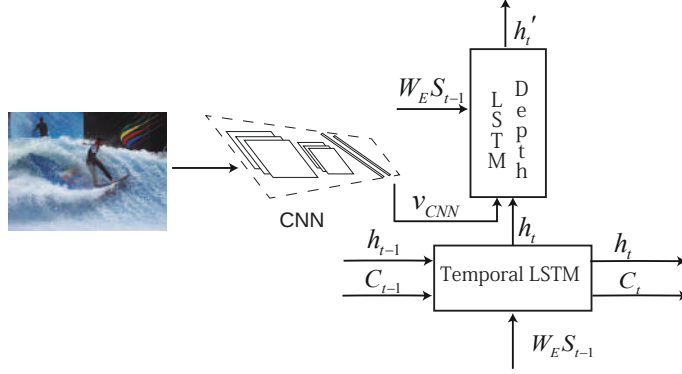


Figure 4.1: An illustration of the proposed model. Our model contains two LSTMs at each step. We use the temporal LSTM to transmit the sequential caption states and the depth LSTM to integrate the visual information.

model will “translate” it into a caption that can be represented as a sequence of words in the same way:

$$(4.4) \quad \mathbf{S} = [\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_N],$$

where  $\mathbf{S}_0$  is a START token added in pre-processing. The objective is to maximize the sum of the log likelihood of the corresponding words:

$$(4.5) \quad \theta^* = \operatorname{argmax}_{\theta} \sum_{t=1}^N \log p(\mathbf{S}_t | \mathbf{I}, \mathbf{S}_0, \dots, \mathbf{S}_{t-1}; \theta),$$

where  $\theta$  represents the parameters to be learned.

The image  $\mathbf{I}$  is encoded as a context vector,  $\mathbf{v}_{CNN}$ , with a Convolutional Neural Network. Then the context vector is injected to an Recurrent Neural Network and the RNN decoder generates a caption word by word. Normally, RNN consists of a sequence of LSTM units. At each step, with the LSTM output, we can obtain the probabilities over the words in the dictionary. Each word is represented by a word embedding vector [56],  $\mathbf{W}_E \mathbf{S}_t$ , where  $\mathbf{W}_E$  denotes a weight metric for the word embedding. The word  $\mathbf{S}_t$  is an indicator one-hot vector [26] in the form of  $1 \times V$ , where  $V$  is the size of the dictionary. It consists of 0 in all the positions, and it is set to 1 if the  $i$ -th position is used to identify a word.

In the conventional encoder-decoder framework, the log likelihood (the conditional probability is simplified as  $p_t$  in Fig. 4.2) is modeled as:

$$(4.6) \quad \log p(\mathbf{S}_t | \mathbf{I}, \mathbf{S}_0, \dots, \mathbf{S}_{t-1}) = f(\mathbf{h}_t),$$

where  $f$  is a nonlinear function that outputs the probability of  $\mathbf{S}_t$ . SoftMax function is exploited here.  $\mathbf{h}_t$  is the hidden state as well as the output in the Recurrent Neural Network. With the LSTM adopted,  $\mathbf{h}_t$  is modeled as:

$$(4.7) \quad \mathbf{h}_t = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{C}_{t-1}),$$

where  $\mathbf{x}_t$  is the input.

### 4.3.2 Recall the Visual Information Continually in the Decoder

In the conventional encoder-decoder framework, the decoder decodes the image features,  $\mathbf{v}_{CNN}$ , at the first time step to provide the initial state for the subsequent time steps. The conventional encoder-decoder framework generates each word only based on the previous state and the previous ground truth word or the previously generated word if scheduled sampling is exploited. The inputs to the LSTM can be represented as:

$$(4.8) \quad \mathbf{x}_t = \begin{cases} \mathbf{v}_{CNN} & t = 0 \\ \mathbf{W}_E \mathbf{S}_{t-1} & t > 0 \end{cases}.$$

Thus, the significant visual information is imported only once to the decoder. Although the LSTM performs well on sequential entities, it still cannot ensure that all the essential information in an image is transmitted completely. Thus, in the conventional encoder-decoder framework, the decoder may lose visual information that is necessary for accurate and information-rich captions and may deviate from the true meaning of the image.

To continuously introduce the visual information into the decoder, the emb-gLSTM [32] concatenates the image features with the word embedding vector and then injects them into the LSTM as extra inputs. The intention is to provide image guidance along the RNN. However, the extra inputs induce additional noise, as indicated in [82], and increase the number of parameters. Thus, the way of importing image representations is crucial, as it should ensure that only relative image information is used to generate the captions.

In our Recall Network, the image information is imported to the decoder in a novel way when generating the captions. Specifically, the visual information is imported to every RNN node by a modified GridLSTM. Thus, the log likelihood is modeled as follows:

$$(4.9) \quad \log p(\mathbf{S}_t | \mathbf{I}, \mathbf{S}_0, \dots, \mathbf{S}_{t-1}) = f(\mathbf{h}'_t),$$

where

$$(4.10) \quad \mathbf{h}'_t = GridLSTM(\mathbf{x}_t, \mathbf{v}_{CNN}, \mathbf{h}_{t-1}, \mathbf{C}_{t-1}).$$

The GridLSTM arranges the LSTM computation in a multidimensional grid. We make use of this property to keep the sequential caption structure and introduce the visual guidance simultaneously. Details on the GridLSTM-based computation are presented in Sec. 4.3.3, including how to compute it and why we use it.

### 4.3.3 Recall with the Depth Dimension LSTM

Intuitively, a generated word should rely on visual information and language constraints adaptively. For example, the image information has little influence when the RNN decoder generates non-visual words such as “of” and “the”. Some generated words obviously rely on the language model such as “phone” after “talking on cell”. In some other situations, the image information is important for the generation, such as recognizing an object as a “dog” or a “man”.

In recent advances [20, 37], the double LSTM design has been demonstrated to be effective for sequence generation. Inspired by [37], rather than concatenating the image features and word embedding features, we use a temporal dimension LSTM (tLSTM) to transmit the sequential caption states and use a depth dimensional LSTM (dLSTM) to integrate the visual information when generating the captions. At the first step in the generation sequence, the visual vector is treated as the tLSTM’s input, and the tLSTM gets word embedding as the input afterwards. At each step, the visual vector is recalled/treated as the previous memory cell in the dLSTM. tLSTM and dLSTM share parameters.

The computation details are presented in Fig. 4.2 (c). The first LSTM (the temporal LSTM), works exactly the same as the conventional LSTM in Equation 7.11. Obtaining the states in the temporal dimension,  $\langle \mathbf{C}_t, \mathbf{h}_t \rangle$ , we further make use of  $\mathbf{h}_t$  in the depth dimension.

On the depth dimension,  $\mathbf{h}_t$  works as the initial hidden state to provide the context information, while  $\mathbf{W}_E \mathbf{S}_{t-1}$  works as the input again to provide the text guidance. The input gate decides to what extent a new subject will be added in, while the forget gate affects the extent to which a previous subject should be forgotten.

$$(4.11) \quad \begin{aligned} \mathbf{i}_t' &= \sigma(\mathbf{U}_i \mathbf{h}_t + \mathbf{W}_i \mathbf{W}_E \mathbf{S}_{t-1} + \mathbf{b}_i), \\ \mathbf{f}_t' &= \sigma(\mathbf{U}_f \mathbf{h}_t + \mathbf{W}_f \mathbf{W}_E \mathbf{S}_{t-1} + \mathbf{b}_f). \end{aligned}$$

The visual information,  $\mathbf{v}_{CNN}$ , is recalled as the previous memory cell in the depth dimension. Through updating the memory cell:

$$(4.12) \quad \mathbf{C}_t' = \mathbf{f}_t' \odot \mathbf{v}_{CNN} + \mathbf{i}_t' \odot \tanh(\mathbf{U}_C \mathbf{h}_t + \mathbf{W}_C \mathbf{W}_E \mathbf{S}_{t-1} + \mathbf{b}_C),$$

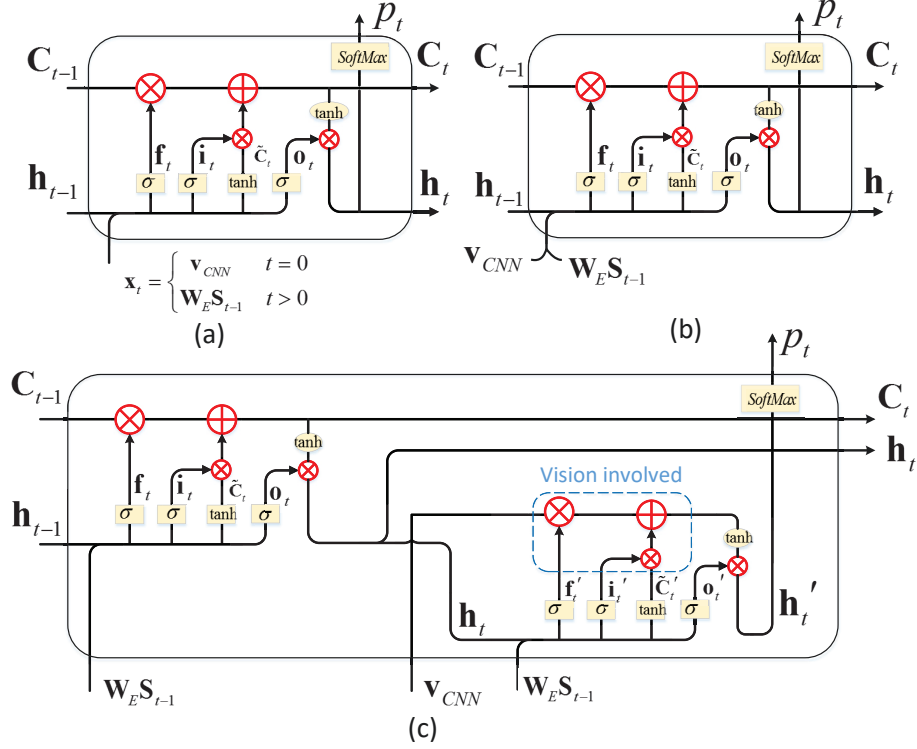


Figure 4.2: Three different decoder units. (a): The image representation is only processed once with the LSTM. (b): The image representation and word embedding are concatenated as extra inputs for the LSTM. (c): Our Recall Network.

some visual information is selectively forgotten through the forget gate that is formulated by the input text,  $\mathbf{W}_E \mathbf{S}_{t-1}$ , and the temporal hidden state,  $\mathbf{h}_t$ . The adaptive visual information is then integrated with the new subject into  $\mathbf{C}_t'$ . Finally, with the output gate, we obtain  $\mathbf{h}_t'$ .

$$(4.13) \quad \begin{aligned} \mathbf{o}_t' &= \sigma(\mathbf{U}_o \mathbf{h}_t + \mathbf{W}_o \mathbf{W}_E \mathbf{S}_{t-1} + \mathbf{b}_o), \\ \mathbf{h}_t' &= \mathbf{o}_t' \odot \tanh(\mathbf{C}_t'). \end{aligned}$$

According to the LSTM interior mechanism [29], the input gate controls the extent to which a new value flows into a cell, and the forget gate controls the extent to which a value remains in a cell. The memory cell interacts with the input gate as well as the forget gate, where it incorporates new information to update and drops certain previous information to forget. In our model, the visual information is recalled as the previous memory cell in the depth dimension. Through updating the memory cell in the depth dimension LSTM, the decoder can respond to both visual and language information

adaptively. While in the conventional encoder-decoder framework, the depth dimension LSTM is not considered, so the image feature remains constant.

With this design, the valuable visual guidance is not confusedly imported into the LSTM with noise but is selectively influenced by the generated token. Additionally, in terms of architecture, our model looks similar to a stacked LSTM [19], as both adopt a hierarchical data structure. Nevertheless, our model considers the generated token and visual information directly in the depth dimension LSTM, which prevents the model from deviating from the accurate image information. Furthermore, we use the inherent functionality of the memory cell in the LSTM to adaptively integrate the visual information. The memory cells facilitate information storage and efficiently code the distributed input within a single cell.

**Computational complexity:** The Recall Network operates efficiently with an update complexity of  $O(W)$  per time step, where  $W$  is the number of parameters. Fig. 4.2 presents the computational details for three different decoder units. Model (a) in Fig. 4.2 corresponds to the conventional encoder-decoder framework, where the image representation is injected only once at the initial step. Model (b) corresponds to the model that takes extra inputs. Model (c) shows the computational details of our Recall Network. Assume that the word embedding is a  $P$ -length vector, while the image representation size and RNN hidden unit size are both  $Q$ . Note that our Recall Network share parameters along the temporal dimension and the depth dimension LSTM. Thus, the parameters to be updated in our model are exactly the same as those in the conventional LSTM:

$$(4.14) \quad W = 4 \times (P \times Q + Q \times Q + Q).$$

In contrast, the number of parameters in the Fig. 4.2 (b) model, which concatenates the image features as extra inputs [32], is  $4 \times ((P + Q) \times Q + Q \times Q + Q)$ . Suppose  $P = Q = 512$ , our model contains approximately 1 million parameters less than the Fig. 4.2 (b) model. Thus, our model contains fewer trainable parameters. One deficiency is that our model may require longer training time due to the computations occurring in two dimensions.

#### 4.3.4 Different Vision Models

For full and dense image captioning, we use different vision models.

**Full Image Captioning:** In full image captioning, a sentence is generated to describe the overall content of the full image. We use a Resnet [28] that is pre-trained on ImageNet [69] to initialize the CNN encoder.

**Dense Captioning:** In addition to the full image captioning, we also investigate our Recall Network for the dense captioning task. Dense captioning requires the model to not only detect the salient regions in an image but also to generate a description for each region, which makes the description specific for each region of attention. It accomplishes the localization and captioning tasks jointly. We use the FCLN model in [35] to obtain the region coordinates and region features in one image.

In the FCLN, a localization layer is inserted between the convolutional layer and the fully connected layer of the VGG-16 [73] network. The localization layer receives an input tensor from the last convolutional layer of the VGG-16 network, identifies the spatial region of the salient objects, and extracts the fixed-length features for each region. The region proposals are regressed from a set of translation-invariant anchors. Given an anchor box  $(x_a, y_a, w_a, h_a)$ , the output region could be presented by the center  $(x, y)$  and shape  $(w, h)$  as follows:

$$\begin{aligned}
 x &= x_a + t_x w_a, \\
 y &= y_a + t_y h_a, \\
 w &= w_a \exp(t_w), \\
 h &= h_a \exp(t_h).
 \end{aligned}
 \tag{4.15}$$

The  $(t_x, t_y, t_w, t_h)$  are the parameters the model needs to learn. To generate a fixed-length feature for each region, and enable the error propagation through the coordinates, a bilinear interpolation is exploited as:

$$V_{C,i,j} = \sum_{i'=1}^W \sum_{j'=1}^H U_{C,i',j'} k(i' - x_{i,j}) k(j' - y_{i,j}),
 \tag{4.16}$$

where  $k$  is the kernel:

$$k(d) = \max(0, 1 - |d|).
 \tag{4.17}$$

$U_{C,i',j'}$  is the feature map in the shape of  $C \times W' \times H'$ , which is generated by the last convolutional layer of the VGG-16 network. The feature,  $V_{C,i',j'}$ , is then fed into a recognition network composed of two fully connected layers, then the final CNN features are fed into our language model. More details about the FCLN can be found in [35].

## 4.4 Experiments

In this work, we propose a novel model, the Recall Network, for image caption generation. The model can recall the visual information continually and import the image



representation through the depth dimension LSTM. To demonstrate the efficiency of our model, experiments are performed on both full image captioning and dense captioning. The elaborated comparison includes the Recall Network vs. the conventional encoder-decoder model, the Recall Network with Reinforcement Learning vs. the conventional encoder-decoder model with Reinforcement Learning, the Recall Network vs. the stacked LSTM, a comparison with the state-of-the-art methods on full image captioning offline and online, and the Recall Network vs. the baseline methods on dense captioning.

#### 4.4.1 Datasets

The datasets used for the experiments consist of images and English descriptions for evaluation. A summary of dataset size is presented in Tab. 4.1.

For the full image captioning, we perform experiments on the MSCOCO [9] dataset, which is provided for the Microsoft COCO caption challenge. Each image is annotated with five descriptive English sentences in the training set and validation set. We perform the offline evaluation on the MSCOCO development set following [81, 99, 102], and perform the online evaluation on the MSCOCO test server. For the development set, we merge all the published annotated data and allocate 5,000 images each for validation and test. The remaining data are used for training.

For the dense captioning, we perform the experiments on the Visual Genome [40] region captions dataset. The Visual Genome dataset contains dense image annotations. It is collected and verified by workers from Amazon Mechanical Turk. The dataset contains 108,077 images in total. On average, there are 40 regions in each image, and each region is annotated with one short sentence. After pre-processing, the images with no region descriptions are removed. We use 5,000 images for validation, 5,000 for testing and 77,398 for training.

Table 4.1: Dataset size summary

Dataset	Train	Test	validation
MSCOCO [9]	82,783	40, 775	40, 504
MSCOCO development set [81]	113,287	5,000	5,000
Visual Genome [40]	77398	5,000	5,000

## 4.4.2 Evaluation Metrics

To evaluate whether the generated description is good, objective evaluation metrics are utilized.

For full image captioning, our experimental results are reported using the MSCOCO caption evaluation tool<sup>1</sup>, including the BLEU [63], METEOR [42], CIDEr [80] and ROUGE-L [46] metrics. BLEU is a popular machine translation metric that analyzes the co-occurrences of n-grams between the candidate and reference sentences, which has good performance for corpus-level comparisons. METEOR is calculated by generating an alignment between the words in the candidate and reference sentences, with the aim of 1:1 correspondence. ROUGE is a set of evaluation metrics designed to evaluate text summarization algorithms. The CIDEr metric measures the consensus in the image captions by performing a Term Frequency Inverse Document Frequency (TF-IDF) weighting for each n-gram. We use the CIDEr metric to choose the models in the validating stage.

Dense captioning is a combination of object detection and image captioning. Following [35], the mean Average Precision (AP) across a range of thresholds for both localization and language accuracy is exploited as the evaluation metric. For localization, we use Intersection over Union (IoU) thresholds of 0.3, 0.4, 0.5, 0.6 and 0.7. For the image description, we use METEOR thresholds of 0, 0.05, 0.1, 0.15, 0.2 and 0.25. Note that there is only one description annotated within a region. We adopt the METEOR since this metric was found to be highly correlated with human judgment in settings with a low number of references [80]. We measure the average precision across all the pairwise settings of these thresholds and report the mean AP.

## 4.4.3 Training Details

**Pre-processing:** For the MSCOCO dataset, we map all the words that occur less than 5 times to the special  $\langle UNK \rangle$  token. For the Visual Genome dataset, we map words that occur less than 15 times to the special  $\langle UNK \rangle$  token. We discard the annotations with more than 10 words, and the images that have fewer than 20 or more than 50 annotations. We even merge the heavily overlapping boxes into a single box.

**Implementation details:** The full image captioning model is implemented using PyTorch. The ResNet101 [28] network, which is pretrained on the ImageNet dataset, is utilized as the image encoder, and no finetuning is conducted. We do not rescale or crop the images. We encode the full image with the final convolutional layer and then

---

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

apply average pooling, which results in a 2048-d vector. In the RNN decoder, both the RNN encoding size and word embedding size are set as 512. The dropout rate is set to 0.5 experimentally. We stop the training after 30 epochs.

The dense captioning model is implemented using Torch. We use the FCLN [35] as the region encoder. The RNN size and word embedding size are set as 512. Following [35], we use five losses in total. A weight of 1.0 is assigned to the cross-entropy loss, while 0.1 is assigned as the weight for other four losses, including the binary logistic losses for region confidence in the localization network and recognition network, and the smooth L1 loss for the region position. Initially, we use a learning rate of  $1 \times 10^{-4}$  for the language model, and a learning rate of  $1 \times 10^{-6}$  for fine tuning the CNN. The Adam [38] algorithm is utilized for stochastic optimization. Our experiments run on an NVIDIA TITAN X GPU. We begin fine tuning the CNN after 100,000 iterations, and stop training after 600,000 iterations.

#### 4.4.4 Experiments on Full Images

For the full image captioning, we present the offline evaluation, including a comparison with baselines and state-of-the-art methods, on the same data split as in [81, 101, 102], and present the online evaluation results on the MSCOCO test server. The Resnet [28] network is utilized as the encoder.

##### 4.4.4.1 The Recall Network vs. the Conventional Encoder-Decoder Model

First, we compare our Recall Network with the baseline, conventional encoder-decoder model (the CEM), to see the effectiveness of our model. The CEM is a variation of the NIC [81] model where we reimplement it with the ResNet as the image encoder. The comparison results are presented in Tab. 4.2. As the beam search can boost the performance by approximately one or two percent, we present the results with greedy sample as well as beam search with size 2. In addition to the basic results, we also show the performance when trained with the reinforcement learning (RL) method [68], which can boost the experiment results over the basic models.

From Tab. 4.2, we can see that our Recall Network outperforms the comparison model in all the metrics. It is worth noting that our Recall Network outperforms the conventional encoder-decoder model by a large margin in the CIDEr metric, where the performance is boosted from 94.50 (99.09 with the beam search) to 98.31 (101.55 with the beam search). Further using the Reinforcement Learning training method

[68], our Recall Network achieves 103.70 in the CIDEr metric. With RL training, our network outperforms the comparison model in all the metrics except METEOR. It is speculated that the METEOR results may be influenced because we use the CIDEr score not METEOR as reward in the RL training.

Table 4.2: Comparison with the baseline methods

		BLEU-4	ROUGE-L	METEOR	CIDEr
CEM	Greedy sample	29.52	52.63	24.68	94.50
	Beam search	31.90	53.55	25.25	99.09
Recall Network	Greedy sample	30.06	53.23	25.28	98.31
	Beam search	<b>32.23</b>	<b>53.90</b>	<b>25.92</b>	<b>101.55</b>
CEM RL	Greedy sample	32.90	54.43	25.01	102.70
	Beam search	33.02	54.48	<b>25.03</b>	102.99
Recall Network RL	Greedy sample	32.91	54.89	24.66	103.41
	Beam search	<b>33.06</b>	<b>54.94</b>	24.67	<b>103.70</b>

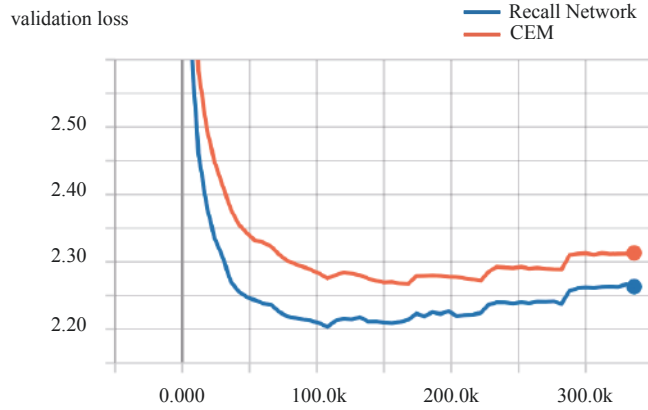


Figure 4.3: Loss comparison on validation split along the training process.

In Fig. 4.3 and Fig. 4.4, we present the loss and the CIDEr value on validation data split during the training process. The validation loss in Fig. 4.3 illustrates that our Recall Network can converge to a lower loss than the conventional encoder-decoder model. Additionally, the CIDEr score in Fig. 4.4 shows that our Recall Network achieves higher CIDEr performance throughout the training process. It can be inferred that our Recall Network can achieve better fitting ability compared with the conventional encoder-decoder model.

In summary, we provide compelling evidence that our Recall Network outperforms the conventional encoder-decoder model.

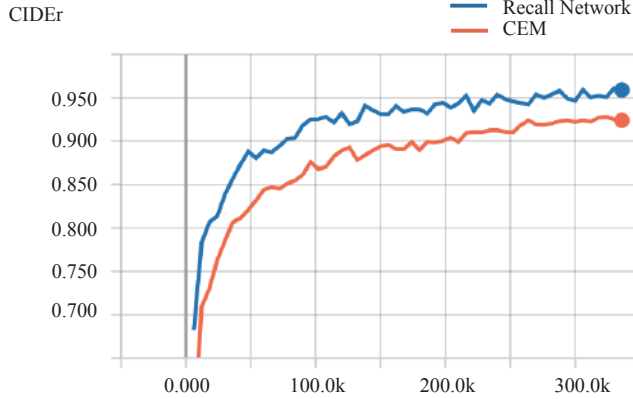


Figure 4.4: CIDEr score comparison on validation split along the training process.

#### 4.4.4.2 The Recall Network vs. the Stacked LSTM

As mentioned before, our Recall Network bears superficial similarity with the stacked LSTM in terms of architecture as both models adopt the hierarchy structure. Differently, our Recall Network involves the visual information and the input token directly in the depth dimension LSTM, while the stacked LSTM only directs the data flow towards another layer. The comparison results for these two models are shown in Tab. 4.3. The stacked LSTM here is a variation of the LRCN[17], where we adopt the ResNet and only inject the image at the start. The stacked LSTM contains two LSTM layers. Our Recall Network significantly outperforms the stacked LSTM in all the metrics, which suggests that the visual and text information directly involved in the depth LSTM provides guidance for the Recall Network. The design of this model can facilitate information storage and prevent the decoder from deviating from the true meaning of the image.

Table 4.3: Comparison with the Stacked LSTM

Models	BLEU-4	ROUGE-L	METEOR	CIDEr
Stacked LSTM	29.12	51.86	24.22	89.58
Recall Network	<b>32.23</b>	<b>53.90</b>	<b>25.92</b>	<b>101.55</b>

#### 4.4.4.3 Comparison with the State-of-the-art Methods

In Tab. 4.4, we compare Recall Network with the state-of-the-art methods. We extract the experiment results from the corresponding papers.

Vinyals *et al.* proposed the widely used CNN-RNN framework, NIC [81]. The LRCN [17] and emb-gLSTM [32] inject image features and the previous word into the LSTM

Table 4.4: Comparison with the state-of-the-art methods on MSCOCO dataset for Offline Evaluation.

Method	BLEU-1	BLEU-4	METEOR	CIDEr
NIC [81]	-	27.7	23.7	85.5
ATT [102]	70.9	30.4	24.3	-
SCN-LSTM [24]	72.8	33.0	25.7	101.2
Full-model [67]	71.3	30.4	25.1	93.7
LSTM-A5 [101]	73	32.5	25.1	98.6
LRCN [17]	62.79	21.00	-	-
DCC [3]	62.79	-	21.00	-
emb-gLSTM [32]	67.0	26.4	22.74	81.25
Bi-LSTM [85]	68.7	25.8	22.9	73.9
Review Net [99]	-	29.0	23.7	88.6
SCA-CNN [8]	71.9	31.1	25.0	-
Soft-Attention [95]	70.7	24.3	23.9	-
Hard-Attention [95]	71.8	25.0	23.04	-
Recall Network	73.40	32.23	<b>25.92</b>	101.55
Recall Network RL	<b>75.82</b>	<b>33.06</b>	24.67	<b>103.70</b>

at each step, where the LRCN [17] adopts a two-layer stacked LSTM. The Bi-LSTM [84][85] generates image captions with a multimodal bidirectional LSTM, and the caption is ultimately decided according to the average word probability within a forward and backward sentence. In DCC [3], the captions are composed by leveraging large object recognition datasets and an external text corpora and by transferring knowledge between semantically similar concepts. The ATT [102] model uses the attribute representation as the semantic attention and combine with image representations for image captioning. The SCN-LSTM [24] makes use of semantic concepts, which is a top-down representation. In the SCA-CNN [8], Review Net [99], Soft-Attention [95] and Hard-Attention [95], the attention mechanism is exploited. The SCA-CNN [8] incorporates spatial and channel-wise attention in a CNN. The Review Net [99] performs a number of review steps with an attention mechanism on the encoder hidden states and injects the outputs to the attention mechanism in the decoder. The Hard-Attention and Soft-Attention correspond to the two alternative attention mechanisms introduced in [95]: stochastic attention and deterministic attention, respectively. The Full-model [67] exploits Reinforcement Learning method with an embedding reward for training. In Table.4.4, the state-of-the-art methods are roughly organize according to the three categories in the abstract.

It should be noted that our Recall Network does not exploit any extra information, such as the attribute representations, and does not utilize an extra net structure, such

as the attention mechanism. As listed in Tab. 4.4, our Recall Network outperforms the methods mentioned above. Specifically, in the CIDEr score, our model is 12.95% higher than the review network [99], which contains extra attentive nets. Compared with the Bi-LSTM [84] that embeds two directional generation results, the single model Recall Network performs better on all listed metrics. Compared with those using external semantic concepts [24, 102], the Recall Network uses only the fully connected features in the CNN but still performs better.

#### 4.4.4.4 Online Evaluation

The MSCOCO team provides a test server that allows people to evaluate their models online. The evaluation is performed on the 40775-image test set. Two evaluation settings are provided: C5 for those with five caption references per image and C40 for those with forty caption references per image. We evaluated our Recall Network, which a single model without any embedding or fine-tuning. The online evaluation results compared with some popular works are presented in Tab. 4.5. Although our model is not the best on the MSCOCO leaderboard, our single model achieves a promising performance that is comparable to the state-of-the-art methods.

Table 4.5: Online Evaluation Performance on the MSCOCO Test Server



Method	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		CIDEr		ROUGE	
	C5	C40	C5	C40	C5	C40	C5	C40	C5	C40	C5	C40	C5	C40
LRCN [17]	71.8	89.5	54.8	80.4	40.9	69.5	30.6	58.5	24.7	33.5	92.1	93.4	52.8	67.8
mRNN [52]	71.6	89.0	54.5	79.8	40.4	68.7	29.9	57.5	24.2	32.5	91.7	93.5	52.1	66.6
ATT [102]	73.1	90.0	56.5	81.5	42.4	70.9	31.6	59.9	25.0	33.5	94.3	95.8	53.5	68.2
Google-NIC [82]	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	94.3	94.6	53.0	68.2
Reviewnet [99]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7	25.6	34.7	96.5	96.9	53.3	68.6
Fukun-Jinjunqi [22]	72.2	90.2	55.6	81.7	41.8	71.1	31.4	60.1	24.8	33.6	93.9	94.6	53.0	68.0
Recall Network	<b>74.9</b>	<b>92.5</b>	<b>58.3</b>	<b>84.2</b>	<b>43.8</b>	<b>73.4</b>	<b>32.1</b>	<b>61.7</b>	24.2	32.1	<b>98.6</b>	<b>102.0</b>	<b>54.3</b>	68.5

#### 4.4.4.5 Qualitative Analysis

To better analyze the practical performance of our model, we present some generated examples in Fig. 4.5. All of the images are from the MSCOCO dataset.

Comparisons between our Recall Network and the conventional encoder-decoder model are presented in the solid box of Fig. 4.5. The CEM somehow fails to give detailed and accurate descriptions. In contrast, the Recall Network can handle some “unseen” image compositions and generate more accurate and detailed captions. For example, our model accurately describes the entity in the fourth image as a “bird” instead of a



				
<b>CEM:</b>	a woman is talking on a cell phone	a man riding a bike down a street	a cat is laying down on a bed	a cat is sitting on a wooden bench
<b>Recall Network:</b>	a woman with glasses is talking on a cell phone	a group of people riding on a street	a cat laying on a bed with a banana	a bird sitting on top of a bowl




				
<b>Recall Network:</b>	a man standing next to a motorcycle	a kitchen with a sink and a window	a bathroom with a toilet and a shower	a woman sitting at a table with a cake
<b>Ground Truth:</b>	<ol style="list-style-type: none"> <li>1. a man with a red helmet on a small moped on a dirt road</li> <li>2. man riding a motor bike on a dirt road on the countryside</li> <li>3. a man riding on the back of a motorcycle</li> <li>4. a dirt path with a young person on a motor bike rests to the foreground of</li> <li>5. a man in a red shirt and a red hat is on a motorcycle on a motorcycle on a hill side</li> </ol>	<ol style="list-style-type: none"> <li>1. a kitchen is shown with a variety of items on the counters</li> <li>2. a kitchen has the windows open and plaid curtains</li> <li>3. a kitchen with two windows and two metal sinks</li> <li>4. an older kitchen with cluttered counter tops but empty sink</li> <li>5. glasses and bottles are placed near a kitchen sink</li> </ol>	<ol style="list-style-type: none"> <li>1. a bathroom with an enclosed shower next to a sink and a toilet</li> <li>2. a clean spacious bathroom with a large shower stall</li> <li>3. there are a toilet a sink and a shower stall in a large bathroom</li> <li>4. a bathroom featuring a walk in shower mirror sink and toilet</li> <li>5. bathroom with a shower sink and toilet in it</li> </ol>	<ol style="list-style-type: none"> <li>1. a young girl inhales with the intent of blowing out a candle</li> <li>2. a young girl is preparing to blow out her candle</li> <li>3. a kid is to blow out the single candle in a bowl of birthday goodness</li> <li>4. girl blowing out the candle on an ice cream</li> <li>5. a little girl is getting ready to blow out a candle on a small dessert</li> </ol>

Figure 4.5: Captions generated by the Recall Network. The comparison between the Recall Network and the conventional encoder-decoder model shows that our model generates more accurate and detailed captions. The comparison between the generated results and the human annotated ground truth shows that further work should be carried out for more comprehensive content captions.

“cat”. Our model captures the “glasses” in the first image when describing the woman, and it captures the object “banana” in the third image. We find that the Recall Network performs better when describing people. One selected example is the second image where it describes multiple people as “a group of people” rather than “a man”.

In the dashed box in Fig. 4.5, we list some suboptimal results as well as the five ground truth descriptions that are annotated by the humans. It is clear that the generated captions are still far from perfect. The generated captions are usually shorter than the human annotated sentences. A further revision can be made by adjusting the maximum sentence length. In addition, the human annotated sentences usually contain



detailed information, e.g., rich adjectives describing the objects, while the generated captions only consist of the name of the major objects. It is inferred that the monotonous image features that are pre-trained for image classification may only be effective for distinguishing the objects. To have the captions contain comprehensive information, e.g. detailed descriptions of the objects, representations with rich visual information are essential. In addition, it can be observed that the language model is overly simplistic when describing indoor scenarios. For instance, it always describes “a kitchen” with “a sink”, and describes “a bathroom” with “a toilet and a shower”. Future research on generating captions with rich content can explore using extra image features and pre-training the language model on the external corpus.

#### 4.4.5 Experiments on Dense Captioning

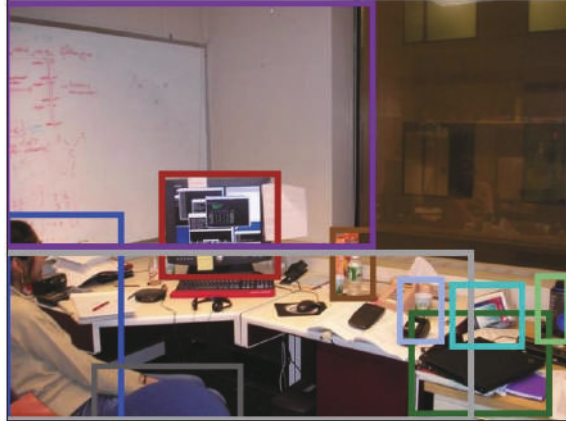
The evaluation results on dense captioning are reported in Tab. 4.6. As mentioned in Sec. 4.4.2, we report the mean Average Precision (AP) across a range of thresholds across detection and captioning. We compare our Recall Network with two baseline models, i.e. FCLN [35] and Recall-Zero Model. For the Recall-Zero Model, we replace the visual representations with all-zero vectors to observe the difference. The FCLN [35] adopted the same decoder as the conventional encoder-decoder model. End-to-end training is conducted in the experiments.

Among the three models, our Recall Network achieves the best performance (6.44 on mean AP), outperforming the baseline FCLN [35] by 1.05 in terms of the AP. In addition, the Recall Network performance is 18% higher than that of the Recall-Zero Model, from which we conclude that the recalled visual information provides significant guidance for the decoder. As both Recall Network and Recall-Zero Model outperform the FCLN, it can be concluded that the GridLSTM structure is effective for text generation. The visualization examples are presented in Fig. 4.6. Compared with the captions generated on the MSCOCO [9] dataset, these captions are shorter and simpler. Attention region detection plays an important role in dense captioning.

As the Recall-Zero Model and Recall Network contain similar architectures, we performed “cross-training” between these two models. After the first round of training, our Recall Network achieved a 6.17 value on AP, which is utilized to initialize the Recall-Zero Model. After fine tuning the Recall-Zero Model for the second round, the Recall-Zero Model achieved a mean AP of 6.26. Afterwards, we used the second round Recall-Zero Model to initialize our Recall Network, and continued the fine-tuning. Finally, our Recall Network performed better than itself in the first round. It seems plausible that the

Table 4.6: Dense Captioning Performance on the Visual Genome Dataset

Methods	FCLN	Recall-Zero Model	Recall Network
AP	5.39	6.26	<b>6.44</b>



a computer monitor. a woman sitting on a table. blue jeans on the chair. black and white laptop. a bottle of bottles on the table. white wall behind the wall . a wooden table. a blue and white bottle. a white plastic bag.

Figure 4.6: Examples generated by the Recall Network in dense captioning. Each caption corresponds to the bounding box in the same color.

“cross-training” among homologous networks can work complementarily and help to boost performance.

## 4.5 Summary

In this chapter, we focus on the language model. We have presented a novel encoder-decoder network, the Recall Network, for generating image-consistent captions. Instead of injecting only the image features at the initial step or using an attention mechanism, we have proposed to recall the visual information continually in the decoder while generating each word, and involve the image features through a depth dimension LSTM. The proposed network is very unique. First, it effectively prevents the network from deviating from the image content, as the image is continually involved in the decoder. Second, the visual information is adaptively admitted through the inherent structure of the LSTM with no extra neural nets or parameters. We conducted exhaustive experiments for full image captioning and dense captioning and provided a comparative evaluation. The experiment results showed that our network outperforms the conventional encoder-

decoder model by a large margin and performs comparably to the state-of-the-art methods. In the future, we plan to extend our model to other multimedia understanding tasks such as text summary.



## VISION MODEL: DUAL GCN

In order to extract rich visual information, in this chapter, we focus on the vision model. We propose to represent image contents via dual level visual graphs. To thoroughly extract visual context information, a region level visual graph and a grid level visual graph are constructed for each image. Though region based attention has been widely applied in captioning, some undetected regions that contain rich background information are omitted. Thus, a grid level graph is introduced to work collaboratively with the region level graph for a comprehensive representation. Graph Convolutional Networks (GCN) are applied to aggregate visual neighborhood information in these graphs.

### 5.1 Introduction

We note that existing generation process is easy to omit rich visual information in the image. In this chapter, we address this issue from the vision aspect. To extract rich visual context knowledge from original images, we represent images with dual level visual graphs, *i.e.* a region-based graph and a grid-based graph. As we know, humans can learn about the characteristics of objects and the relationships that occur between them to reason a large variety of visual concepts. Compared with the traditional CNN raw image encoding, graph representation of an image can naturally incorporate this kind of reasoning ability over the visual context. Recently, there have been efforts to utilise region-based attention/GCN [2, 100] for image captioning. However, these methods fail to explore the rich background visual context due to the fact that detection algorithms

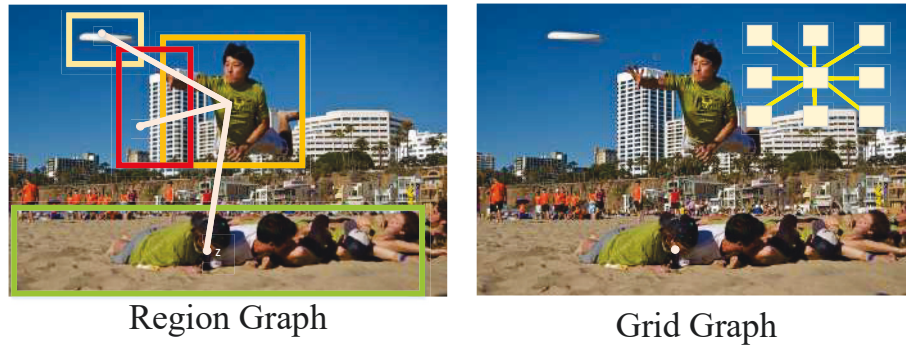


Figure 5.1: Region-level and grid-level visual graph construction.

tend to only consider salient areas in an image. Thus, in addition to a region level visual graph, a novel grid level visual graph is adopted as an essential supplement. The grid level graph is constructed on a full image’s feature map with designed labels, which can encode fine-grained background context ignored by regional objects. As each instance in the feature map can correspond to a visual receptive field in the raw image, this visual graph can connect grid neighbors in the full image. An illustration is shown in Fig. 5.1. Above the dual level visual graphs, we employ Graph Convolutional Networks (GCN) [39, 53] to encode images by learning high-order correlations among graph nodes.

The contribution of this chapter can be summarized as follows. To thoroughly explore visual context, we represent an image with a region-level graph and a grid-level graph. GCN is utilized to integrate the visual neighborhood information among graph nodes.

## 5.2 GCN Background

Graph Neural Networks [70] were introduced to combine graph structure data with neural networks. Recently, GCN which extend CNN to aggregate information from graph structure data have received increasing research attention. There are two streams of GCN construction: spectral GCN [16, 39] and spatial GCN [7, 10, 58]. Spectral GCN is based on the spectrum of graph Laplacian. Features of the graph are firstly transformed using Fourier transform and the convolutions are performed in the spectral domain. Spatial GCN approaches define convolutions directly on general graphs, operating on spatially close neighbors. Our method falls into this category. A number of method exploited GCN in computer vision applications. In [97], a spatial-temporal GCN was exploited for human action recognition. Brown *et al.* [60] applied GCN for visual question

answering and learned a graph structure automatically. In [53], a directed graph was constructed for sentence encoding which can bring more information than the undirected graphs. However, there is only one work [100] applies GCN to image captioning. Different from [100] which contains a spatial graph and a semantic graph for global context, we construct two labeled directed graphs spatially. These two graphs extract visual context on both grid level and region level.

### 5.3 Image captioning by two visual relation graphs

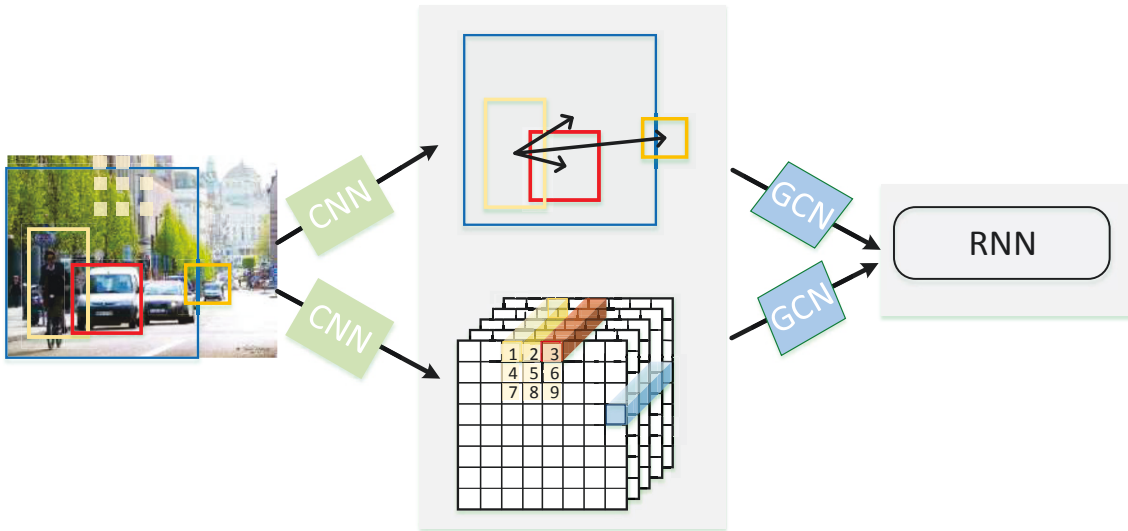


Figure 5.2: The proposed CNN-GCN-RNN model. The image is represented by a region level graph and a grid level graph. GCN is utilized to encode neighbourhood information.

In this section, we begin by briefly describing our CNN-GCN-RNN captioning model as shown in Fig. 5.2. We construct visual graphs from two aspects: (1) a region graph of the current image where regions in the image are nodes and spatial relationships between these regions are edges; (2) a grid graph where we represent feature map instances as nodes and build edges to encode detailed context ignored by regional object representation. Then, GCN is employed to integrate neighborhood information in the graph.

### 5.3.1 Problem Formulation

Formally, a captioning system receives an image  $\mathbf{I}$  as the input and is required to output a sentence  $S$  to describe the image content.  $\mathbf{S}$  can be represented as a sequence of words,  $\mathbf{S} = \{\mathbf{w}_1, \mathbf{w}_2 \cdots \mathbf{w}_N\}$ , where  $\mathbf{w}_t$  denotes the  $t$ -th word. During training,  $(\mathbf{I}, \mathbf{S})$  is given as a training example pair, and the image caption model (parameterized by  $\theta$ ) can be optimized by minimizing the cross entropy loss:

$$(5.1) \quad L(\theta) = -\sum_{t=1}^N \log p(\mathbf{w}_t | \mathbf{I}, \mathbf{w}_0, \cdots, \mathbf{w}_{t-1}; \theta).$$

The image caption model is typically achieved by the CNN-RNN framework, *i.e.*, the image  $I$  is encoded by a CNN, and a RNN decodes the internal image feature into a sequence of words.

Inspired by recent success of GCNs [39, 53, 100], in our approach, we formulate our basic model as a CNN-GCN-RNN structure as shown in Fig. 5.2. We represent the internal image feature by leveraging two visual relationship directed graphs: a region level graph  $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$  and a grid level graph  $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$ .  $\mathcal{V}$  and  $\mathcal{E}$  denote node sets and edge sets respectively. Noteworthily, different nodes and edges are established in these two graphs. Then, we fuse these two graphs before generation.

### 5.3.2 Region Visual Graph

Firstly, we use Faster R-CNN [66] to detect salient image regions and objects. Region of interest (RoI) pooling is utilized to extract the region level feature for each object. Instead of train such a model from scratch, we utilize the pretrained Faster R-CNN bottom-up attention feature in [2]. As in [10, 100], the graph  $\mathcal{G}_r$  is constructed on the detected regions based on their spatial relationship. We regard each image region as a node  $v$  and represent the node by the region level feature. For  $\mathcal{E}_r$ , the edges are established according to spatial relationships between every two regions, and the edge are labeled with the manually designed class numbers in [100]. Specifically, class 1 “inside” and class 2 “cover” are established if object  $o_i$  is fully covered with object  $o_j$ . Class 3 “overlap” is established if the intersection over union (IoU) between  $o_i$  and  $o_j$  are larger than 0.5. Then we can compute the ratio  $\alpha_{ij}$  between the relative distance and the diagonal length of the whole image, and the relative angle  $\delta_{ij}$  between two bounding boxes. Class 4-11 can be established as  $\lceil \delta_{ij}/45^\circ \rceil + 3$  and  $\text{IoU} < 0.5$ . Otherwise, no edge is established.



### 5.3.3 Grid Visual Graph

Region level graph embeds a bunch of objects feature considering the region neighbourhood. However, some image background are excluded from these object regions due to the fact that detection algorithms only consider salient areas in the image. Only region graph cannot cover all the context in the image. Thus, a grid level graph is necessary as the supplement for region graph. In this chapter, we propose a grid level graph which is constructed on the full image feature map.

Since each instance in the feature map can correspond to a visual receptive field in the raw image, we construct a graph on the grid feature map. Adjacent nodes on the features map corresponds to neighboring regions in the full image. We first use a CNN to extract the full image convolutional feature. For example, we use Resnet 101 and get a feature map sized  $14 \times 14 \times 2048$ . Then we treat each instance on the feature map as a node and construct edges among the 196 instances. Edges are established within a 8-connected neighborhood. Edge labels are class number 1~9 as shown in Fig. 5.2.

### 5.3.4 GCN for Directed Labeled Graph

With the constructed graph, we apply GCN to integrate neighborhood information of each node in the graph. We can get a high-order representation by considering the connection among graph nodes. In this work, in order to fully incorporate directions and labels in the graph, we exploit a variation of GCN proposed in [53]:

$$(5.2) \quad \mathbf{v}_i^{k+1} = ReLU \left( \sum_{j \in \mathcal{N}(i)} \mathbf{W}_{dir(i,j)}^k \mathbf{v}_j^k + \mathbf{b}_{lab(i,j)}^k \right),$$

where  $k$  denotes the layer of neighbors we consider.  $\mathcal{N}(i)$  represents the set of neighbors of node  $v_i$ .  $dir(i,j)$  includes three types of edges, *i.e.*, the edge from  $i$  to  $j$ , the edge from  $j$  to  $i$  and the self loop edge. The edge labels are explicitly encoded in the bias vector  $\mathbf{b}_{lab(i,j)}$ , and  $lab(i,j)$  indicates the edge label as describe before. Parameters are not shared among different edges and labels.

Rather than uniformly accepting information from all neighboring nodes, a scalar gating mechanism is utilised:

$$(5.3) \quad g_{i,j}^k = \sigma \left( \hat{\mathbf{W}}_{dir(i,j)}^k \mathbf{v}_j^k + \hat{\mathbf{b}}_{lab(i,j)}^k \right),$$

where  $\sigma$  is the logistic sigmoid function.  $\hat{\mathbf{W}}_{dir(i,j)}$  and  $\hat{\mathbf{b}}_{lab(i,j)}$  are the weight and bias for the gate. With the gating mechanism, the final computation is formulated as:

$$(5.4) \quad \mathbf{v}_i^{k+1} = ReLU \left( \sum_{j \in \mathcal{N}(i)} g_{i,j}^k \left( \mathbf{W}_{dir(i,j)}^k \mathbf{v}_j^k + \mathbf{b}_{lab(i,j)}^k \right) \right).$$

In this way, new node representation  $\mathbf{v}_i^{k+1}$  can integrate information from neighbouring nodes as well as labeled edges. The quantitative effect for the dual graph caption generators is experimentally studied in Table 5.1.

### 5.3.5 RNN Generator

Next we introduce the basic RNN generator. As the region level graph  $\mathcal{G}_r$  and grid level graph  $\mathcal{G}_g$  contain different amounts of nodes, we cannot simply concatenate the node embedding after separately applying GCNs over  $\mathcal{G}_r$  and  $\mathcal{G}_g$ . We instead combine two updated graphs into a matrix  $m \times d$ , where  $m$  are the number of nodes in dual graphs, and  $d$  denotes the dimension of the node representation. Then we feed this visual matrix into the RNN generator.

In the RNN generator, we exploit the top-down attention mechanism [2], which contains an attention LSTM (A-LSTM) and a language LSTM (L-LSTM) at each step. The input for the attention LSTM consists of the previous output of the language LSTM, the mean-pooled visual feature  $\bar{\mathbf{v}} = \frac{1}{m} \sum_m \mathbf{v}_i$  and the previously generated word  $\mathbf{w}_t$ :

$$(5.5) \quad \mathbf{h}_t^1 = lstm([\mathbf{h}_{t-1}^2, \bar{\mathbf{v}}, \mathbf{W}_E \mathbf{w}_t]),$$

where  $\mathbf{W}_E$  is a word embedding matrix.

Given the output  $\mathbf{h}_t^1$ , the visual feature with attention is calculated as:

$$(5.6) \quad \begin{aligned} a_{i,t} &= \mathbf{W}_a \tanh(\mathbf{W}_{va} \mathbf{v}_i^k + \mathbf{W}_{ha} \mathbf{h}_t^1), \\ \lambda_t &= softmax(\mathbf{a}_t), \\ \mathbf{e}_t &= \sum_{i=1}^m \lambda_{i,t} \mathbf{v}_i^k. \end{aligned}$$

Then, we can get the output of the language LSTM:

$$(5.7) \quad \mathbf{h}_t^2 = lstm([\mathbf{e}_t, \mathbf{h}_t^1]).$$

With the output  $\mathbf{h}_t^2$ , we can get the predicted word  $\mathbf{w}_{t+1}$  through a softmax layer.

## 5.4 Experiments

### 5.4.1 Experimental Setup

**Dataset:** We train and evaluate our model on MSCOCO [9], which is the most popular captioning benchmark used for Microsoft COCO caption challenge. MSCOCO contains

82,783 training images, 40,504 validation images and 40,775 unlabeled testing images. Each image in the training set and validation set is annotated with five English descriptive sentences. For offline evaluation, we adopt a widely used Karpathy’s data split [68, 99], where 5K images are used for validation, 5K for testing and 113,287 for training. For caption processing, following [68, 100], we convert all descriptions to lower case and map words that occur less than five times to  $\langle UNK \rangle$  tokens.

**Evaluation Metrics:** To evaluate the quality of the generated description, objective evaluation metrics are utilized given the human annotated ground truth. Results are reported with metrics: BLEU [63], METEOR [42], ROUGE-L [46], CIDEr [80] and SPICE [1]. The evaluation metrics are provided by MSCOCO caption evaluation tool<sup>1</sup>. In the self-critic training, we use CIDEr score as the reward. In the validation stage, we also use CIDEr to choose the best model.

**Implementation Details:** In the visual encoder, we apply faster R-CNN [66] in cooperating with ResNet101 [28]. For the region level GCN, 36 regions with top detection confidences are utilized. Each region is represented as a 2,048-dimension feature vector. We directly use the feature trained with bottom-up attention [2]. For the grid level GCN, we encode the image with the final convolutional layer in ResNet101 and apply spatial adaptive pooling, which results in a  $14 \times 14 \times 2048$  feature map. We construct the grid graph on the feature map. Thus, the grid level graph contains 196 nodes, and each node is represented as a 2,048-dimension feature vector. All the input features are well pre-trained with finetuning. The node representation after GCN is still set as 2,048.

In the RNN captioning model, we set the hidden state size in LSTM as 1,024 and the word encoding as 512. The hidden size for measuring attention distribution is set as 512. Dropout is set as 0.5 experimentally.

We start training the basic captioning model using Adam optimizer with an initial learning rate of  $5 \times e^{-4}$ . Scheduled sampling and learning rate decay start from the beginning. After 30 epochs training, we start the self-critic training [68] and stop at the 70-th epoch. For the self-critic training, the initial learning rate remains  $5 \times e^{-4}$ . We run the experiments on an NVIDIA TITAN X GPU.

### 5.4.2 Comparative Models

**Ablative models:** To study the key components of our method, we implement several variants of our model. (i) ResNet baseline encodes images with ResNet101 and decodes it

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

with the top-down attention RNN [2]. (ii) Grid GCN constructs a grid level graph on the ResNet feature map. (iii) Region GCN baseline is the region level GCN with top-down attention RNN as [100]. (iv) RegionGrid GCN refers to the dual level GCN vision model. **State-of-the-arts:** The performance of our model is compared with that of the state-of-the-art models, whose performance are extracted from corresponding papers. (i) GCN-LSTM [100] fuses a spatial GCN and a semantic GCN. GCN-LSTM-rl refers to the model trained with self-critics. (ii) RFNet [33] uses multiple CNN encoders and a fusion network between the encoder and the decoder. (iii) SCST is the first to use self-critic training [68], and is trained with soft attention mechanism [95]. (iv) SR-PL [49] uses self-critic training as well as a text-to-image retrieval reward. (vi) Up-Down [2] exploits a combination of bottom-up and Top-down attention. (vii) Based on scene graphs, Obj-R+Rel-A [45] applies visual object features and semantic relationship features for representation. Obj-R+Rel-A-rl refers to the model trained with self-critics. (viii) SGAE [98] incorporates a scene graph auto-encoder into the conventional CNN-RNN framework.

Table 5.1: GCN Model Performance comparisons on MSCOCO offline set

Model	cross entropy loss								Self critic training							
	Greedy sample				Beam search				Greedy sample				Beam search			
	C ↑	M ↑	B4 ↑	B1 ↑	C ↑	M ↑	B4 ↑	B1 ↑	C ↑	M ↑	B4 ↑	B1 ↑	C ↑	M ↑	B4 ↑	B1 ↑
ResNet baseline	105.1	26.4	32.3	74.7	107.8	26.8	33.9	74.8	117.5	27.4	35.7	78.4	120.1	27.6	36.1	78.8
Grid GCN	109.0	26.7	33.0	75.6	112.9	27.5	35.2	76.3	121.5	27.8	36.8	79.1	122.3	27.8	36.8	79.3
Region GCN baseline	113.5	27.3	34.9	77.1	115.3	27.8	36.7	76.9	125.3	28.1	37.9	80.0	125.4	28.1	37.9	80.0
RegionGrid GCN	115.2	27.5	35.4	77.5	117.6	28.0	37.1	77.6	125.8	28.5	38.0	80.1	126.1	28.5	38.2	80.4

### 5.4.3 Ablative Analysis

Table 5.1 shows the performance of ablative models on MSCOCO offline split. We present performance from two optimizing method: cross-entropy-loss training and self-critic [68] training. Besides, performance from two decoding strategies are reported: using greedy sample and beam search.

**Evaluation for the Grid level graph.** (1) *ResNet baseline v.s. Grid GCN.* The GridGCN is constructed on the ResNet feature map. It is clearly that the GridGCN achieves a significant improvement over the ResNet baseline. (2) *Region GCN baseline v.s. Region-Grid GCN.* The Region GCN is similar to GCN-LSTMspa[100]. Noteworthily, the batch size is set as 1,024 and training epoch is more than 200 in [100], which is quite large compared to most published methods and is beyond our computation resources. For fair comparison, we implement the GCN-LSTMspa with batch size as 60 and max epoch as 70, and we use the same setting in our all comparative experiments. With the consistent

setting, it can be seen that the RegionGrid GCN outperforms the implemented Region GCN.

**Evaluation for Grid graph updating iterations.** To see the effect of the iterations when updating the Grid graph, we conduct competitive experiments in Table.5.2. Grid GCN-itr1 refers the model updating once. Grid GCN-itr2 refers to the one updating twice. It can be seen that the Grid GCN updating twice can achieve the best performance.

Table 5.2: Comparisons on the updating times of the Grid GCN.

Model variants	Greedy sample				Beam search			
	C	M	B4	B1	C	M	B4	B1
Resnet	105.1	26.4	32.3	74.7	107.8	26.8	33.9	74.8
Grid GCN-itr1	107.0	26.5	32.1	75.1	108.3	26.8	33.7	75.1
Grid GCN-itr2	<b>109.0</b>	<b>26.7</b>	<b>33.0</b>	<b>75.6</b>	<b>112.9</b>	<b>27.5</b>	<b>35.2</b>	<b>76.3</b>
Grid GCN-itr3	106.0	26.4	32.3	74.9	108.6	26.8	33.8	75.6

#### 5.4.4 Comparison with State-of-the-art Models

**Offline evaluation.** The offline comparison with state-of-the-art models is presented in Table 5.3. All comparative models are evaluated in the commonly used Karpathy’s data split. We present the results into two categories based on cross entropy loss training and self-critic training. Overall, the evaluation results optimized by cross entropy loss generally indicate that our model achieves superior performance against other state-of-

Table 5.3: Performance comparisons with the state-of-the-art methods on MSCOCO offline set

-	B1 ↑	B4 ↑	M ↑	R ↑	C ↑	S ↑
UP-Down [2]	77.2	36.2	27.0	56.4	113.5	20.3
RFNet [33]	76.4	35.8	27.4	56.5	112.5	20.5
GCN-LSTM [100]	77.4	37.1	28.1	57.2	117.1	21.1
Obj-R+Rel-A [45]	76.7	33.8	26.2	54.9	110.3	19.8
Dual-GCN (ours)	77.6	37.1	28.0	57.2	117.6	21.1
SCST [68]	-	34.2	26.7	55.7	114.0	-
SR-PL [49]	80.1	35.8	27.4	57.0	117.1	21.0
Up-Down-rl[2]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet-rl[33]	79.1	36.5	27.7	57.3	121.9	21.2
GCN-LSTM-rl[100]	80.9	38.3	28.6	58.5	128.7	22.1
SGAE[98]	80.8	38.4	28.4	58.6	127.8	22.1
Obj-R+Rel-A-rl [45]	79.2	36.3	27.6	56.8	120.2	21.4
Dual-GCN-rl (ours)	80.4	38.2	28.5	58.2	126.1	21.8

the-art methods. Specifically, our model achieves 117.6 on the CIDEr score. In self-critic training, our performance is slightly lower than GCN-LSTM-rl. As we mentioned before, our batch size and training time are much less than those in GCN-LSTM-rl [100]. With the similar experiment setting, our model outperforms other latest models such as Up-Down-rl, RFNet-rl, SCST, SR-PL.

**Online evaluation.** We also compare our model with the state-of-the-arts using the MSCOCO online test server. Table 5.4 reports the performance on testing images with five (c5) and forty (c40) reference captions. We include performing methods that have been officially published. Though not being the best one on the MSCOCO Learderboard, our single model achieves a promising performance among many ensemble models.

Table 5.4: Performance evaluation on MSCOCO online Test Server

Method	B1		B2		B3		B4		M		C		R	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Google-NIC [82]	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	94.3	94.6	53.0	68.2
Reviewnet [99]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7	25.6	34.7	96.5	96.9	53.3	68.6
Adaptive [51]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	104.2	105.9	55.0	70.5
SCST [68]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	114.7	116.7	56.3	70.7
Up-Down [2]	80.2	95.2	64.1	88.1	49.1	79.4	36.9	68.5	27.6	36.7	117.9	120.5	57.1	72.4
Dual-GCN (ours)	79.8	94.5	64.2	88.4	49.5	79.2	37.4	68.4	28.4	37.4	121.5	122.9	58.0	72.8

## 5.5 Summary

In this chapter, we have proposed dual level visual graphs to extract rich visual context. We develop a region visual graph as well as a grid visual graph for each image, and GCN is utilized to integrate neighborhood information on the labeled directed graphs. Experiments on MSCOCO indicate that our model outperforms comparative baselines and achieves a promising result.

## TRAINING STRATEGY: NOISE AGENT

In addition to vision models and language models, we note the training strategy also has impact on the captioning performance. In this chapter, a trainable noise module is introduced in the generator to pursue captions' diversity and accuracy. Specifically, we inject an additive noise in the Recurrent Neural Networks (RNN) for perturbation and train the noise module towards a better evaluation metric. Regarding the noise module as an agent and generating noise as an action, we train this noise module by a variant of REINFORCE algorithm, where noiseless results are viewed as baselines. Experiment results on MSCOCO dataset show that our model outperforms comparative models and can achieve promising results on diversity and accuracy.

### 6.1 Introduction

Diversity in descriptions is an essential property in human language, since different people tend to describe the same thing with different sentences due to their various language preference. However, existing captioning models, especially those trained with maximum likelihood objectives, end up to generate common plain sentences with small vocabulary. This is primarily due to the deficiencies in the generated word distribution and generator's strong bias towards frequent fragments in the training samples. Though recently proposed Variational Auto-Encoder (VAE) and Generative Adversarial Net (GAN) have been exploited for caption diversity [15, 88], they tend to generate captions with accuracy tradeoff. The generated sentences are plain and with limited variability



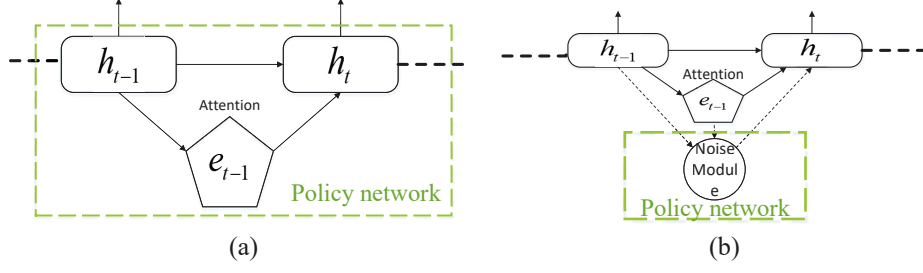


Figure 6.1: (a) Existing RL framework in captioning. (b) Our proposed one with the noise module.

in expression. Thus, it is important to explore a method for diverse generation without losing the accuracy.

In this chapter, we boost the caption diversity by adding a trainable noise module in the RNN generator. Previous work has investigated the effects of noise in Recurrent Neural Networks. In [34], noise injection to network parameters has been demonstrated to improve generalization and convergence performance. In [64], noise was added to agent parameters for more consistent exploration and a rich set of behaviours in Reinforcement Learning. In [11], Cho proposed to add unstructured noise to transition hidden states as small perturbation in the hidden space corresponds to jumping from one plausible configuration to another. In [25], the unstructured noise was replaced by a policy network with deterministic action. In this work, we propose to inject adaptive noise in the hidden states depending on the current states, attention and visual inputs. We use this noise to add perturbation in the hidden state space, which results in diverse generations.

To train such a module, we view the noise module as an agent with a stochastic gaussian policy in RL and regard generating noise as an action. The other network such as the basic CNN-RNN framework is fixed and regarded as part of the environment. We approximate the gaussian policy with a parametric function. As shown in Fig. 6.1 (b), this design is significantly different from existing RL caption model, where the CNN-RNN network is generally viewed as a policy network and the action is to predict the next word given conditional probabilities over the vocabularies. To approximate the gaussian noise policy, we alter a variant of REINFORCE algorithm (SCST[68]). Specifically, we view the noiseless output as a baseline to normalize the noised-added generation rewards. It inherits the advantages of REINFORCE as it is capable of directly optimizing the evaluation metric, and it further avoids to construct a baseline network for the noise. In this way, noise from the module that gets higher-reward return than noiseless generation



will be “pushed up”, while the interiors will be suppressed.

In this work, we propose a novel model, G-noise, for image captioning. We first train the basic CNN-GCN-RNN network, and then fix the basic module and train the plug-in noise module. We evaluate our method on MSCOCO dataset [9]. Exhaustive experiment results demonstrate that our model is able to generate more accurate and diverse captions and further achieve promising performance. Main contributions of this chapter can be summarized as follows:

- To address caption diversity, we introduce a plug-in noise agent between RNN hidden states. We regard the noise function as a stochastic gaussian policy and generating noise as an action.
- While training the noise module in REINFORCE algorithm, noiseless reward is employed as the baseline. This noise-critic design encourages promising noise samples and avoids to construct a baseline network.

## 6.2 RL Background

Deep reinforcement learning is used in a wide range of sequential decision making problems [57, 72]. The standard Reinforcement Learning (RL) [76] framework consists of an agent interacting with an environment, executing a series actions and aiming to maximize the cumulative rewards. Recently, several attempts have been made to apply Reinforcement Learning, especially the policy gradient method, to image captioning and sequence generation tasks. Generally, the recurrent model is viewed as an agent where the action is to predict the next word. The parameters of the generator define a policy and the reward can be any evaluation metrics.

In [65], Ranzato *et al.* first introduced RL into an RNN-based sequence model, which mixed together the cross-entropy loss and the REINFORCE objective in training process. After that, Liu *et al.* [48] proposed to replace the mixed training with Monte Carlo rollouts, and they used the policy gradient method to optimize a combination of two NLP metrics. Rennie *et al.* [68] used the classical REINFORCE [91] algorithm and proposed a novel baseline obtained by the current model. This self-critic training method is easy and efficient while maximizing the evaluation metric CIDEr [80]. Besides, Actor-Critic method was introduced to image captioning in [67], where a policy network and a value network worked collaboratively to generate captions. Although previous works have adopted RL in captioning, there is no model specifically devised for caption diversity and

accuracy. In this work we introduce a noise agent in the RNN generator. Significantly different from existing RL framework, we regard the noise module as an agent and generating noise as the action. Parameters in the noise agent define a policy, while the pretrained CNN-GCN-RNN model is viewed as part of the environment. To maximize the reward metrics, we introduce a variant of self-critic training.

### 6.3 Boosting Caption Diversity with a Noise Agent

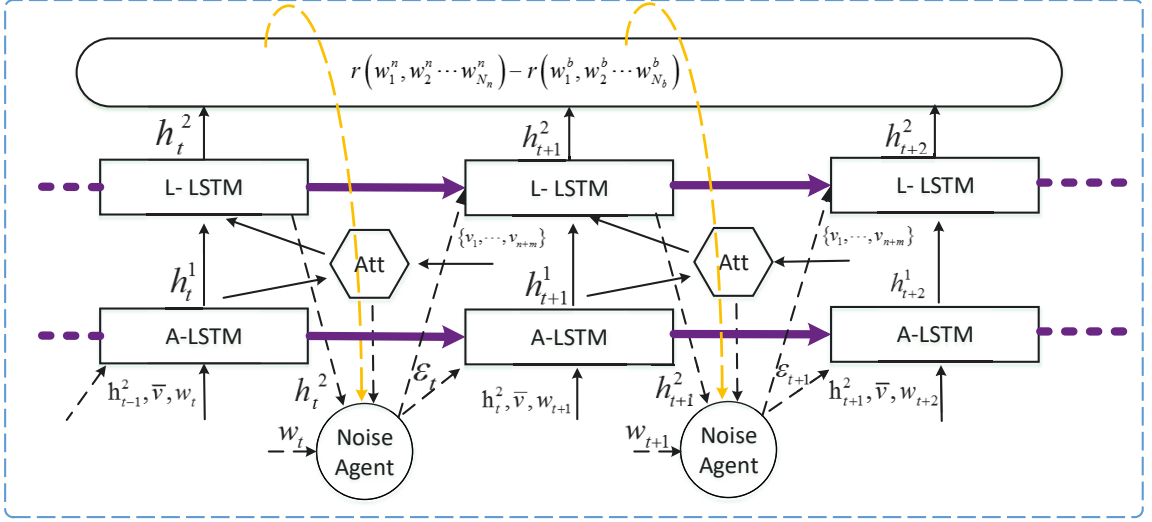


Figure 6.2: The transition details in the noise-added model.

#### 6.3.1 Gaussian Noise Agent

In previous works [11, 25, 34, 64], the effects of noise in RNN have been studied by injecting noise to the parameter space or the transition hidden states. We drew inspiration from recent advances in neural machine translation [11, 25]. In [11], a translation with a higher log-probability was found by injecting unstructured noise into RNN hidden states. In [25], the unstructured noise was replaced by a trainable parametric function. From these work, we can see that small perturbation in the hidden space results in jumping to plausible distributions in the word space.

In our work, we inject an adaptive noise in the RNN hidden states to achieve perturbation in current word distribution space. We propose a trainable Gaussian noise module

to manipulate the language LSTM hidden states. The noise module receives the input as previous language LSTM hidden state  $\mathbf{h}_{t-1}^2$ , previously generated word  $\mathbf{w}_t$  and the visual attention vector  $\mathbf{e}_t$ . The noise module outputs a vector  $\varepsilon_t$  which can be added to  $\mathbf{h}_{t-1}^2$ .

$$(6.1) \quad \begin{aligned} \hat{\mathbf{h}}_{t-1}^2 &= \mathbf{h}_{t-1}^2 + \varepsilon_t, \\ \pi_\phi : \varepsilon_t &\sim \mathcal{N}(\mu, \sigma^2), \end{aligned}$$

where  $\mu$  and  $\sigma$  denotes the mean and standard deviation of the distribution.  $\mu$  and  $\sigma$  can be modeled by an MLP given the inputs aforementioned, where tanh and sigmoid are utilized as the activation function respectively. With reference to [11, 43], uncertainty is often greatest when predicting earlier symbols and gradually decreases as more and more context becomes available for the conditional distribution. Cognitively, injecting noises to the early hidden states contributes more to the diversity and grammar of the sentence than it does to the later ones, because more language context or constraints are available along the decoding process. Thus, we inject scheduled noises to the hidden states:

$$(6.2) \quad \hat{h}_{t-1}^2 = h_{t-1}^2 + \frac{\varepsilon_t}{t}.$$

Additionally, we clamp noises into the range  $(-\gamma, \gamma)$  to avoid inferior perturbation. The adaptive noise contributes to sentence accuracy compared to naive noise. In Fig. 6.3, we show caption examples generated with naive noise and the adaptive noise. It can be seen that the adaptive noise results in more reasonable generation, while the naive noise may breach the grammar.

In order to train the noise module, we propose a new RL framework. The noise module is viewed as the agent in RL. The action is set to generate noise from the gaussian distribution. We regard others, such as the original RNN model, visual representation and words, as part of the environment. Fixing the basic model, we only approximate the Gaussian noise policy  $\pi_\phi$ . The evaluation metric (*e.g.*, CIDEr) is used as the episode reward. A detailed illustration can be seen in Fig. 6.2.

### 6.3.2 Noise Critic Training

Recently, rather than use the cross entropy loss, some works directly optimize the NLP metrics with Reinforcement Learning technics. In existing RL captioning framework, RNN is viewed as an agent that interacts with the environment. The environment



Naïve noise: a man and a woman holding a white snowboard **and a black and white photo**

Adaptive noise: a man and a woman holding a snowboard standing next to each other



Naïve noise: a large group of airplanes on a cloudy day **with an airplane**

Adaptive noise: a view of an airport terminal with planes parked in it

Figure 6.3: Generation examples with naive noise and adaptive noise.

contains the visual features and predicted words. Parameters in the RNN define a stochastic policy which predicts the next word given the probability in Equation 5.1. After executing an action, the agent receives a scalar reward. The training goal is to minimize the negative expected rewards (*e.g.*, the NLP evaluation metric):

$$(6.3) \quad L(\theta) = -\mathbb{E}_{S \sim \pi_\theta} \left[ \sum_t r_{w_t} \right].$$

In order to get the gradient  $\Delta_\theta L(\theta)$ , some policy gradient methods are exploited, *e.g.* REINFORCE [91]. REINFORCE uses Monte Carlo sample to get the episode reward, *i.e.* play out the whole episode to compute the total reward  $r_S$ . The gradient can be computed as:

$$(6.4) \quad \begin{aligned} \Delta_\theta L(\theta) &= -\mathbb{E}[r_S \Delta_\theta \log p_\theta(S)], \\ &\approx -r_S \Delta_\theta \log p_\theta(S). \end{aligned}$$

In this way, the gradients encourage the parameters increase in the direction proportional to the reward, which makes actions with high rewards more likely. Normally, a baseline is involved to reduce variance. The baseline can be any function, even a random variable. An alternative gradient can be computed as:

$$(6.5) \quad \Delta_\theta L(\theta) \approx -(r_S - b) \Delta_\theta \log p_\theta(S).$$

In our work, to approximate the gaussian noise policy, we alter a variation of REINFORCE algorithm, SCST [68]. SCST optimizes the captioning generator following the conventional RL framework, and it propose an effective baseline. SCST use the output of greedy sample generation as the baseline. Thus, sampled results that outperform the greedy sampled ones can get positive return.

In our task, we use the basic generation reward (noiseless output) as the baseline. The gradients in the noise module can be calculated as:

$$(6.6) \quad \Delta_{\phi} L(\phi) \approx -(r_{S_n} - r_{S_b}) \Delta_{\phi} \log p_{\phi}(\varepsilon),$$

where  $r_{S_n}$  denotes the CIDEr reward for noise-augmented generation, and  $r_{S_b}$  is the reward for the basic (noiseless) generation. As a result, the sampled noise that get a higher return than the noiseless output can be encouraged, but inferior noise can be suppressed. In Gaussian distribution, the log probability of the generated noise is:

$$(6.7) \quad \log p_{\phi}(\varepsilon) = -\frac{(\varepsilon - \mu)^2}{2\sigma^2} - \log \sigma - \log \sqrt{2\pi}.$$

In this design, we avoid constructing a baseline network or a critic network (in actor-critic method). We train the noise policy by comparing noise augmented results and basic results. Better perturbation will be pushed up, while inferior will be suppressed.

### 6.3.3 Parallel Noise Decoding

In the training stage, we pretrain the basic model and then train the noise agent. In the inference stage, we exploit a parallel decoding scheme. Specifically, We run  $M$  noise augmented decoding in parallel, and output the sequence with the highest log probability accumulation. This is a way to generate a stable noise augmented caption. The effects of noise agent is experimentally demonstrated in Table 6.1, Table 6.4 and Fig. 6.5.

## 6.4 Experiments

### 6.4.1 Experimental Setup

We use the CNN-GCN-RNN model in last chapter as the basic model. Experiment details can be found in Chapter 5. The MLP in noise agent has a hidden layer sized 32. Noise threshold  $\gamma$  is set to 1 experimentally. Given the pre-trained basic model, we train the noise agent for extra 30 epochs with initial learning rate at  $8 \times e^{-5}$ . In the inference stage, 5 parallel decoding processes are conducted. In the inference stage, 5 parallel decoding processes are conducted. We run the experiments on an NVIDIA TITAN X GPU.

### 6.4.2 Comparative Models

**Ablative models:** To study the key components of our method, we implement several variants of our model. (i) BottomUp baseline encodes images with Bottom-up attention

[2] and decodes it with the soft-attention LSTM [95]. (ii) BottomUp + noise agent uses the ButtonUp baseline as the basic model and augments with the trainable noise module. (vi) RegionGrid GCN refers to the dual level GCN vision model introduced in Chapter 5. (vii) RegionGrid GCN + noise agent refers to the proposed model.

Table 6.1: Performance comparisons for ablation study on MSCOCO offline set

Model	cross entropy loss								Self critic training							
	Greedy sample				Beam search				Greedy sample				Beam search			
	C ↑	M ↑	B4 ↑	B1 ↑	C ↑	M ↑	B4 ↑	B1 ↑	C ↑	M ↑	B4 ↑	B1 ↑	C ↑	M ↑	B4 ↑	B1 ↑
BottomUp baseline	107.9	26.6	33.8	76.5	112.0	27.3	36.0	77.0	121.5	27.7	36.7	79.1	121.7	27.7	36.8	79.2
BottomUp + noise agent	111.0	27.1	34.6	76.5	112.8	27.4	36.5	76.5	123.1	27.8	37.0	79.4	123.7	27.9	37.0	79.4
RegionGrid GCN	115.2	27.5	35.4	77.5	117.6	28.0	37.1	77.6	125.8	28.5	38.0	80.1	126.1	28.5	38.2	80.4
RegionGrid GCN + noise agent	116.4	27.8	35.6	77.2	118.2	28.0	37.3	77.5	125.8	28.2	37.8	80.2	126.1	28.1	37.9	80.1

**State-of-the-arts:** The performance of our model is compared with that of the state-of-the-art models, whose performance are extracted from corresponding papers. (i) GCN-LSTM [100] fuses a spatial GCN and a semantic GCN. GCN-LSTM-rl refers to the model trained with self-critics. (ii) RFNet [33] use multiple CNN encoders and a fusion network between the encoder and the decoder. (iii) SCST is the first to use self-critic training [68], and is trained with soft attention mechanism [95]. (iv) SR-PL [49] uses self-critic training as well as a text-to-image retrieval reward. (vi) Up-Down [2] exploits a combination of bottom-up and Top-down attention.

### 6.4.3 Ablative Analysis

Table 6.1 shows the performance of ablative models on MSCOCO offline split. We present performance from two optimizing method: cross-entropy-loss training and self-critic [68] training. Besides, performance from two decoding strategies are reported: using greedy sample and beam search.

**Evaluation for the noise module.** (1) *BottomUp baseline v.s. BottomUp + noise agent.* The results using different optimizing methods and different decoding strategies show that the noise augmented model consistently outperforms the baseline model. In Fig. 6.4, the CIDEr and BLEU-4 performance log of the noise module on validation set are depicted. It can be seen that our noise-critic optimizing is effective, where the model jumps out to a rough performance first and then converges to a new value. Practically, we find that injecting noise to soft attention results in a more stable metric log than injecting to the TopDown attention. It may because TopDown has two layer LSTMs and different layers may exhibit different sensitivities to the noise. (2) *RegionGrid GCN v.s. RegionGrid GCN + noise agent.* It shows that the noised model generally outperforms

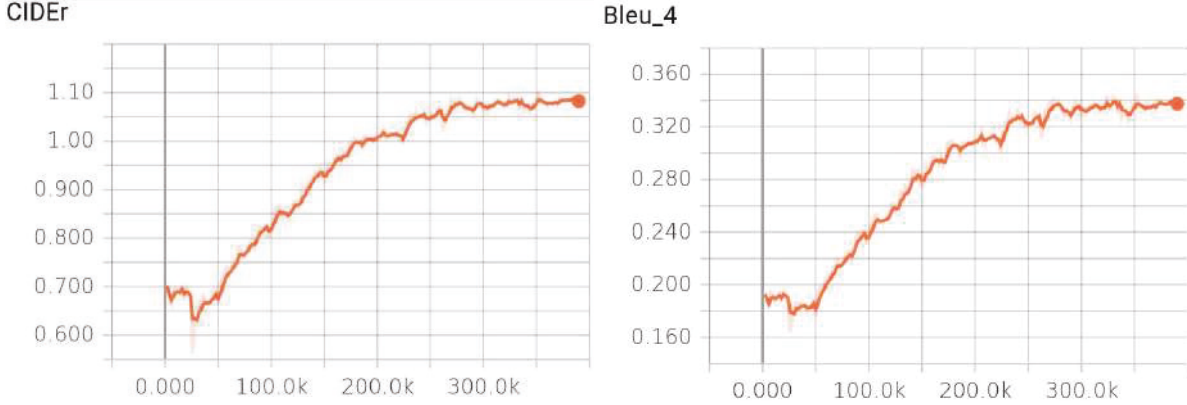


Figure 6.4: CIDEr and BLEU-4 log on validation set while training the noise agent

Table 6.2: Performance comparisons with the state-of-the-art methods on MSCOCO offline set

-	B1 ↑	B4 ↑	M ↑	R ↑	C ↑	S ↑
UP-Down [2]	77.2	36.2	27.0	56.4	113.5	20.3
RFNet [33]	76.4	35.8	27.4	56.5	112.5	20.5
GCN-LSTM [100]	77.4	37.1	28.1	57.2	117.1	21.1
G-noise (ours)	<b>77.5</b>	<b>37.3</b>	28.0	<b>57.3</b>	<b>118.2</b>	<b>21.3</b>
SCST [68]	-	34.2	26.7	55.7	114.0	-
SR-PL [49]	<u>80.1</u>	35.8	27.4	57.0	117.1	21.0
Up-Down-rl[2]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet-rl	79.1	36.5	27.7	57.3	121.9	21.2
GCN-LSTM-rl[100]	<b>80.9</b>	<b>38.3</b>	<b>28.6</b>	<b>58.5</b>	<b>128.7</b>	<b>22.1</b>
G-noise-rl (ours)	<u>80.1</u>	<u>37.9</u>	<u>28.1</u>	<u>58.1</u>	<u>126.1</u>	<u>21.8</u>

the RegionGrid GCN baseline, where CIDEr score is improved from 115.2 to 116.4 with greedy sampling and cross entropy loss.

#### 6.4.4 Comparison with State-of-the-art Models

**Offline evaluation.** The offline comparison with state-of-the-art models is presented in Table 6.2. All comparative models are evaluated in the commonly used Karpathy’s data split. We present the results into two categories based on cross entropy loss training and self-critic training. Overall, the evaluation results optimized by cross entropy loss generally indicate that our model achieves superior performance against other state-of-the-art methods. Specifically, our model achieves 118.2 on the CIDEr score. In self-critic training, our performance is slightly lower than GCN-LSTM-rl. As we mentioned before,



our batch size and training time are much less than those in GCN-LSTM-rl [100]. With the similar experiment setting, our model outperforms other latest models such as Up-Down-rl, RFNet-rl, SCST, SR-PL.

**Online evaluation.** We also compare our model with the state-of-the-arts using the MSCOCO online test server. Table 6.3 reports the performance on testing images with five (c5) and forty (c40) reference captions. We include performing methods that have been officially published. Though not being the best one on the MSCOCO Leaderboard, our single model achieves a promising performance among many ensemble models.

Table 6.3: Performance evaluation on MSCOCO online Test Server

Method	B1		B2		B3		B4		M		C		R	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Google-NIC [82]	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	94.3	94.6	53.0	68.2
Reviewnet [99]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7	25.6	34.7	96.5	96.9	53.3	68.6
Adaptive [51]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	104.2	105.9	55.0	70.5
SCST [68]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	114.7	116.7	56.3	70.7
Up-Down [2]	80.2	95.2	64.1	88.1	49.1	79.4	36.9	68.5	27.6	36.7	117.9	120.5	57.1	72.4
G-noise (ours)	79.8	94.3	64.0	88.0	49.2	78.5	37.2	67.5	28.0	36.8	121.2	123.4	57.7	72.3

### 6.4.5 Diversity Evaluation

To evaluate the corpus level diversity, two evaluation criteria are reported in Table 6.4 similarly to [71]. (1) Vocabulary size: the number of unique words in all generations. (2) % Novel sentences: the percentage of all generated sentences not seen in the training split. We conduct this experiment on the 5K test images. It shows that the noise module results in a diversity improvement over the baseline.

Table 6.4: Diversity Evaluation

-	Vocabulary $\uparrow$	%Novel Sentences $\uparrow$
BottomUp baseline	838	72.6
BottomUp + noise agent	888	73.5

### 6.4.6 Qualitative analysis

Fig. 6.5 shows the qualitative examples of generated captions. Three kinds of captions are presented: caption produced by ResNet baseline, caption produced by our G-noise model and one of human annotated ground truth. From the visualized results, it is easy to see that our G-noise captions tend to exhibit a more diverse sentence structures. We



highlight the more descriptive fragments in G-noise captions with the red color that the baseline captions do not have. These fragments somehow make the captions accurate through enriching the description. For instance, in the last image, compared to the simple sentence structure “a man is in the air with a frisbee”, our model depicts the image with a man “jumping” to “catch” a frisbee ” on the beach”.

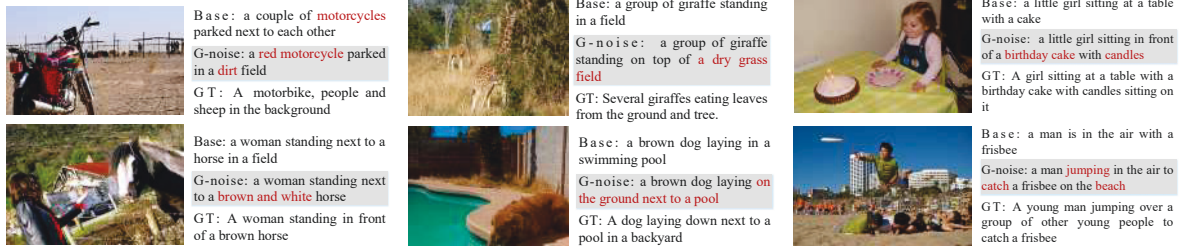


Figure 6.5: Generated captions from ResNet baseline, our G-noise model and the Ground Truth (GT).

## 6.5 Summary

In this chapter, we have proposed a novel model, G-noise generator, to pursue accuracy and diversity in image captioning. Unlike previous CNN-RNN framework, our model contains an additive noise module which can manipulate the transition hidden states in the RNN decoder. To train such a module, we regard the noise module as an agent with a stochastic gaussian policy, and generating noise is regarded as the action. The noise module is trained by an introduced noise-critic training algorithm, where we use noiseless results as the baselines in the REINFORCE algorithm. This design can involve adaptive perturbation to boost caption diversity. We believe the perturbation can somehow make the optimization algorithm jump out the local minimization. Experiments on MSCOCO indicate that our model outperforms comparative baselines and achieves a promising result towards diversity and accuracy.



# **Part III**

## **Extensions**



## CREATING MODERN CHINESE POETRY FROM IMAGES

Artificial creativity has attracted increasing research attention in the field of multimedia and artificial intelligence. Despite the promising work on poetry/painting/music generation, creating modern Chinese poetry from images, which can significantly enrich the functionality of photo-sharing platforms, has rarely been explored. In this chapter, we extend image captioning to the image-to-poetry task. Existing generation models cannot tackle three challenges in this task: (1) Maintaining semantic consistency between images and poems. (2) Preventing topic drift in the generation. (3) Avoidance of certain words appearing frequently. These three points are even common challenges in other sequence generation tasks. In this chapter, we propose a Constrained Topic-Aware Model (CTAM) to create modern Chinese poetries from images regarding the challenges above. Without image-poetry paired dataset, we construct a visual semantic vector to embed visual contents via image captions. For the topic-drift problem, we propose a topic-aware poetry generation model. Additionally, we design an Anti-Frequency Decoding (AFD) scheme to constrain high-frequency characters in the generation. Experiment results show that our model achieves promising performance, and is effective in poetry's readability and semantic consistency.

### 7.1 Introduction

Recently, artificial creativity has attracted increasing research attention in the field of multimedia understanding and artificial intelligence [14, 27, 83, 92]. Researchers have

been exploring machine capability in human-level creative products [50, 74, 104, 106] like poetry, story, music, painting, etc. Poetry is regarded as an advanced form in linguistic communication as its expressive form in aesthetics and semantics. In this chapter, we focus on creating modern Chinese poetry with visual inspirations, which has rarely been investigated. Our work can be applied to various scenarios, e.g., generating personalized poems for photo storage/sharing platforms like Google Photo, Instagram, etc. It can also enrich the function of chatbot system on phone/tablet/PC terminals. Moreover, automatically creating modern Chinese poetry from images is a challenging task in artificial creativity, which interacts with computer vision as well as natural language processing.

Existing works on automatic poetry generation are mostly rule-based approaches or statistical ones. The rule-based approaches[59, 61, 77, 93] focus on the form of words, characters, rhythms and templates. Netzer et al.[59] proposed to generate Japanese poetry Haiku from a seed word using association norms. The algorithm in [61] created Portuguese poems by filling in sentence templates. However, the rule-based approaches are inappropriate for our task as modern Chinese poetry is mostly written in free verse which does not follow consistent patterns. The statistical approaches[89, 90, 103, 104] learn to generate poems by extracting statistical patterns in the existing poetry corpus. In [104], classical Chinese poetry can be generated by a Recurrent Neural Network language model. In [89], Wang et al. proposed to create classical Chinese Song iambics with an attention-based sequence-to-sequence model, in which the first sentence should be provided as a cue sentence. Even though these algorithms show promising results in poetry generation, few effort has been attempted on the modern Chinese poetry, which is significantly different from the classical Chinese poetry. Importantly, these algorithms cannot explore the semantic consistency between poems and images. Moreover, existing algorithms fail to deal with a practical problem, i.e., the frequent occurrence of certain words. In fact, some words like “I”, “you”, “one” and “dream” occupy a large ratio in the modern Chinese poetry corpus. The imbalanced training data make themselves frequently appear in the generation results, which may undermine the readability of generated poems. Therefore, it is critical and essential to explore a method to convey the visual semantics and tackle the high-frequency-word problem for poetry generation.

Recently, some keyword-based algorithms [90, 94, 96] are introduced to handle semantic topics in sequence generation. In [90], four keywords are arranged in an order manually for classical Chinese poetry, and each keyword is assigned with a line as a subtopic. In [94], Xing et al. proposed to generate interesting responses for chatbots

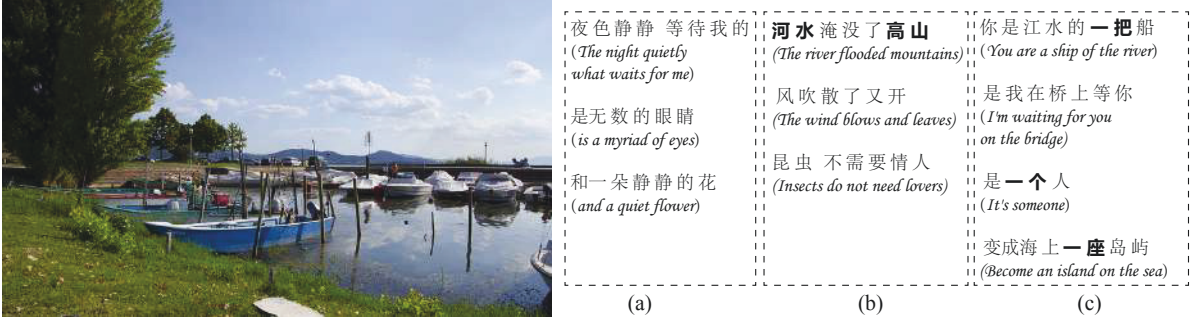


Figure 7.1: Three challenges in the image-poetry generation. (a) Semantic inconsistency between the image and the poetry. (b) Topic drift in the subsequent lines. (Relative words are shown in bold.) (c) Frequent occurrence of certain words (shown in bold).

by incorporating topic words in the sequence model [75]. Therefore, it is reasonable to convey the visual semantics with topic keywords in the poetry generation. However, existing algorithms still cannot deal with the following challenges in our task: (1) The generation may perform semantic inconsistency with the given image. As shown in Fig. 7.1(a), the generated example is defective due to the irrelevance with the image. (2) The generation may suffer the topic-drift problem. The “topic drift” is defined as the scenario that the generation deviates from the given topic gradually. In other words, the generator “forgets” the visual semantics gradually. One example is shown in Fig. 1 (b), where only the first line is related to the visual semantics. We refer that in the conventional sequence model, such as variations of Seq2Seq [75], the encoded vector is only used to provide initial states, thus the generation may deviate from the topic induced in the first line. (3) The generation may suffer the high-frequency-word problem as mentioned above in modern Chinese Poetry, which undermines the readability. The poem shown in Fig. 7.1(c) performs poor readability as the word “One” (shown in bold) repeatedly appears among four lines. These challenges are worth investigating since they are common and practical in natural language generation.

In this chapter, we identify three major challenges in image-to-modern-Chinese-poetry creation. To tackle these three challenges, we propose a Constrained Topic-Aware Model (CTAM). To ensure the semantic consistency between images and poems, we utilize an image caption model as a bridge. A visual semantic vector is constructed with key components in the caption, and it serves as the poetry topic. To deal with the topic-drift problem, we propose a topic-aware poetry generation model, which contains two LSTMs at each step. We use a temporal LSTM to transmit sequential Chinese characters, and

use a depth LSTM to recall the semantic vector selectively. To constrain the occurrence of high-frequency characters, we propose an Anti-Frequency Decoding (AFD) scheme based on Mutual Information (MI). The pipeline of our approach is briefly illustrated in Fig. 7.2. There does not exist any image-poetry (modern Chinese poetry) dataset for end-to-end training. Matching images with poems is diversely subjective. Thus, we exploit image captioning as a bridge to convey visual semantics. The image description is translated into a Chinese sentence, from which a semantic vector is formed as the poetry topic. Captioning model is typically optimized with cross-entropy loss, which cannot correlate well with human assessments. To generate captions with rich semantics like human creating, we apply the captioning model augmented by Reinforcement Learning [65, 68, 91] in this task. Then, through the topic-aware poetry generation model and anti-frequency decoding, an appropriate poem can be created. In the experiment, the qualitative evaluation and quantitative evaluation clearly demonstrate that our approach can effectively perform semantic consistency and proper readability.

The main contributions of our work can be summarized as:

- This is a pioneer study on modern Chinese poetry creation from visual information. We figure out three major challenges in this task: semantic inconsistency, topic drift and word re-appearance.
- We propose a Constrained Topic-Aware Model for image-to-poetry creation, where we introduce visual semantic vectors to ensure semantic consistency between images and poems, we propose a topic-aware poetry generation model to recall visual topics selectively and prevent the topic-drift problem, and we propose the Anti-Frequency Decoding scheme based on MI to constrain the occurrence of particular characters.
- Qualitative and quantitative experiments demonstrate the effectiveness of our approach, and it leads to dramatic improvement on readability and semantic consistency.

The rest of the chapter is structured as follows. In Section 7.2, some previous works on poetry generation are reviewed. In section 7.3, we describe our approach in details. The experiment comes to the Section 7.4.



## 7.2 Background

Poetry generation is an interesting task in intelligent computational creativity. A variety of approaches have been proposed on this topic, while most of them are inspired by text information rather than visual semantics. Existing work on poetry generation can be roughly divided as rule-based approaches and statistical approaches.

The rule-based approaches generally focus on the form of words or characters, and they generate poems based on patterns or templates. [59, 77, 93] were proposed to generate Japanese poetry, Haiku/Renku, by searching phrases which match the syntactic patterns and theme words. The algorithm in [77] started with a user-supplied seed word and generated poems based on syntactic constraints. The algorithm in [59] generated Haiku from a seed word using association norms. However, all of them depend on the sensitivity of the seed phrase. Oliveira [61] proposed to generate Portuguese poetry by filling in sentence templates. The algorithm in [13] also used templates to construct poems according to constraints on rhyme, meter, stress, sentiment, word frequency and word similarity. These approaches can benefit user-interactive generation. However, the rule-based approach is not suitable for modern Chinese poetry generation, as free verse is the dominant form. A fixed template is not available for modern Chinese poetry.

More recently, statistical approaches received more research attention by constructing a language model to extract statistical patterns in an existing poetry corpus. In [104], given keywords, Chinese quatrains can be generated by a Recurrent Neural Network language model. The first line was generated based on keywords, while the subsequent lines were generated based on all previously generated lines. In [89], Wang et al. applied an attention-based sequence-to-sequence model to generate Chinese Song iambics, in which the first sentence should be provided as a cue sentence. These two approaches share the limitation that the subsequent lines may deviate from the semantic topic inducted at the first line. To address it, Zhang et al. [103] proposed a memory-augmented neural network to consider innovation in Chinese classical poems. In [90], to generate quatrains, a keyword is assigned to each line. The problem is that the keyword orders have to be set manually, which limits the poem flexibility and results in incoherency across lines. In [31], a phonetic-level neural network model and a constrained character-level neural network model were proposed for rhythmic poetry in a variety of forms in English.

Recently, some works generated styled descriptions of images, which is related to image-poetry generation somehow. In StyleNet [23], Gan et al. proposed to generate

humorous or romantic descriptions for images using a standard image-caption dataset and an external monolingual text dataset. In SentiCap [54], Mathews et al. proposed to generate captions with positive or negative sentiments. This is achieved by two labeled datasets: a standard image-caption dataset and a dataset containing captions with sentiments. This supervised transforming method is interesting, but it is expensive and difficult to scale up.

Different from the above-mentioned methods, in this work, we generate modern Chinese poems from images. This work directly connects visual semantics with modern Chinese poetry generation. The existing work [96] generated Chinese quatrains from images using a memory-augmented network. However, an image-poetry paired dataset must be provided. Additionally, it ignored the semantic gaps between image objects and classical Chinese poetry contents. In [47], Liu et al. proposed to generate English free verse from images in an end-to-end fashion. While, a large paired dataset should be collected and annotated by human annotators. In our work, there is no image-poetry(modern Chinese poetry) dataset. We utilize image caption as a bridge and keep the semantic between images and poetries consistently. Moreover, due to the variant form of modern Chinese poetry, constructing a language model is more challenging in our work.

### 7.3 Creating Poetries From Images

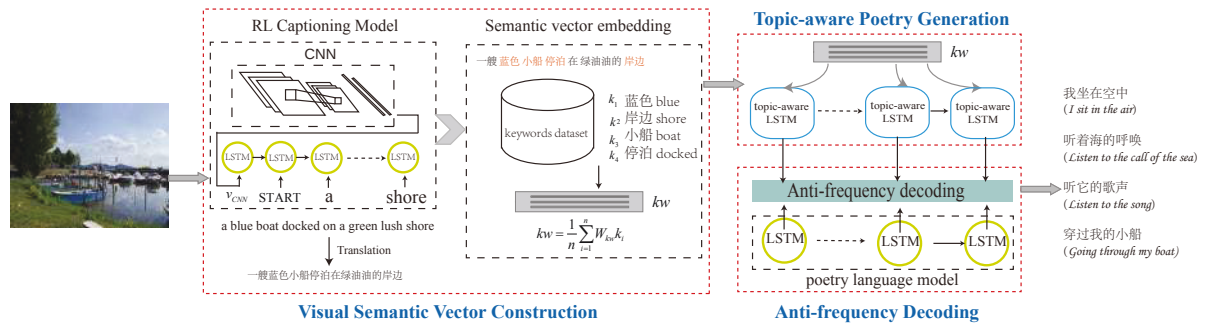


Figure 7.2: The framework of our proposed CTAM

The framework of our approach is depicted in Fig. 7.2, which contains three major steps: visual semantic vector construction, topic-aware poetry generation, and anti-frequency decoding. In this section, we will first formulate the problem and then introduce the details of our approach.

Table 7.1: Listing of notations in this chapter

Notation	Description
$I$	the input image
$W = [w_0, w_1 \cdots w_N]$	the generated poem
$C = [c_0, c_1 \cdots c_M]$	the generated image caption
$k_i$	a keyword in the translated caption
$kw$	the semantic vector
$G = (V, E)$	text graph in the TextRank algorithm
$WS(V_i)$	the TextRank score for vertex $i$
$w_{ji}$	weight of an edge in the text graph
$d$	a damping factor in the TextRank algorithm
$h_t$	a hidden state in RNN
$i_t, f_t, o_t$	input/forget/output gates in LSTM
$C_t$	a memory cell in LSTM
$v_t$	the frequency of the character $w_t$ in corpus
$s_t$	the decoding score
$\alpha$	the frequency threshold in the decoding scheme
$\lambda$	the penalty hyperparameter in decoding scheme
$W_{kw}$	the keyword embedding matrix to be learnt
$U_{i/f/o/C}, W_{i/f/o/C}$	the parameters in LSTM to be learnt

### 7.3.1 Problem Formulation

Given an image  $I$ , our goal is to create a modern Chinese poem  $W$  which can be represented as a sequence of Chinese characters  $W = [w_0, w_1 \cdots w_N]$ . To achieve this, we construct a generation model for  $p(w_t|I, w_0, \cdots w_{t-1})$ .

We use the image caption  $C = [c_0, c_1 \cdots c_M]$  as a bridge between images and poems, which encodes visual semantics and provides the topics for the generation. To present the visual semantics, a semantic vector  $kw$  is constructed by keywords  $k_1, k_2 \cdots k_n$  in an English-Chinese translated caption. Then, the semantic vector  $kw$  is injected into our proposed topic-aware poetry generation model, which can deal with the topic-drift problem in the Recurrent Neural Network. With topic-aware LSTMs, we get  $p(w_t|kw, w_0, \cdots w_{t-1})$ . Lastly, in decoding, an anti-frequency decoding score,  $s_t$ , is proposed to inhibit the frequently appearing characters such as “I”, “Love”. We list key notations in Tab.7.1.

### 7.3.2 Visual Semantic Vector Construction

In order to create poetries related to the given image, we need to mine semantic information from the visual image first. As there is no available image-poetry paired dataset,

we exploit the image caption as a bridge in this work, and then we construct a semantic vector through keywords in the caption.

### 7.3.2.1 Captioning Model

Image captioning is an appropriate bridge connecting computer vision and natural language processing. Different from traditional computer tasks such as object recognition that only recognizes major objects in an image, image captioning can generate a natural language sentence to describe the full image. A caption contains ample semantic information including objects, adjectives, verbs even prepositions.

We use a CNN-RNN model [81] to generate captions. The given Image  $I$  is encoded with a Convolutional Neural Network (CNN) as  $v_{CNN}$ . The encoded features are then decoded into a sequence of words by a Recurrent Neural Network (RNN). The objective function is typically to minimize the cross-entropy loss

$$(7.1) \quad L(\theta) = - \sum_{t=1}^M \log p(c_t | v_{CNN}, c_0 \cdots c_{t-1}; \theta),$$

where  $\theta$  are the parameters to be learnt.

However, the typical objective function cannot correlate well with human assessments. The non-differentiable metric CIDEr [80] can measure the captioning quality using human consensus. Recently, a Reinforcement Learning (RL) algorithm, REINFORCE [91], is introduced in the recurrent neural network to deal with the gradient of the non-differentiable function [65]. It can optimize the gradient of the expected reward (the CIDEr score in this work) by sampling from the model during training. To get captions with rich semantic information like human creating, we exploit an RL captioning model in this task. A self-critic algorithm [68] is exploited as it avoids estimating the reward signal through the self-critics. The alternative objective function is to minimize the negative expected reward, and the expected gradient of the reward function can be computed as follows:

$$(7.2) \quad L(\theta) = -\mathbb{E}_{c^s \sim p_\theta} r(c_0^s, c_1^s \cdots c_M^s),$$

$$(7.3) \quad \Delta_\theta L(\theta) \approx - (r(C^s) - r(C^g)) \Delta_\theta \log p_\theta(C^s),$$

where  $r(C^s)$  is the reward of the caption sampled from the model, and  $r(C^g)$  is the reward of the caption obtained by greedy sampling.

Since there is no available large-scale Chinese image-caption paired dataset, we use MSCOCO [9] dataset to train the captioning model, and then translate the English descriptions into Chinese with Baidu Translation API<sup>1</sup>. Different from classical Chinese poetry, modern Chinese poetry is mostly written with modern vernacular, which can be easier to correspond to the translated captions. Generally, the captioning model trained on MSCOCO can describe most of the images.

### 7.3.2.2 Semantic Vector Embedding

We utilize keywords in the translated caption to form a visual semantic vector. Firstly, we collect a poetry keyword dataset. Then, the Chinese caption is segmented into words or phrases. Relative words/phrases in the caption are selected as visual keywords by retrieving in the poetry keywords dataset. At last, the visual keywords are represented as word embedding vectors and form a semantic vector.

To construct the poetry keyword collection firstly, we extract keywords from each poem by an unsupervised method TextRank [55]. The TextRank algorithm is a graph-based ranking algorithm to evaluate the importance of words. Text is represented as a graph  $G = (V, E)$ , and words can be added as vertices in the graph. Edges are added between two words based on the co-occurrence. The TextRank score is iterated as:

$$(7.4) \quad WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j),$$

where  $w_{ji}$  is the weight of the edge, and  $d$  is a damping factor that is set to 0.85 in our experiment. The algorithm is iterated till converge, and the vertices are sorted based on their scores. Words with higher scores are selected as keywords in the corresponding text.

In Fig. 7.3, we show keywords obtained from two different visual models. It is clear that the caption provides rich information, while object recognition only provides three object names. It is reasonable to exploit the captioning model for visual information interpretation.

Admittedly, some styled image descriptions [23, 54] are more attractive and contain more emotional words. However, in our approach, we use image-caption data to achieve visual consistency, and then we train a generation model on poem corpus to achieve poetic imagination. Intuitively, we would like to avoid the biases towards a large number

<sup>1</sup><https://fanyi.baidu.com/>



Figure 7.3: Comparison on keywords obtained from two different visual models.

of emotional topic words in our task. For instance, given an image with a *moon*, we prefer the poem related to the *moon* rather than only related to *loneliness*. We argue that the basic adjectives and verbs generated by our captioning model contain sentiments somehow.

To find the proper visual keywords, we segment the caption into words, then retrieve each word in the poetry keyword dataset. In order to encode the visual semantic information, we construct a semantic vector by the selected visual keywords. A pre-trained word2vec [56] model is utilized then. Each keyword is represented as a word embedding vector, and the semantic representation is formed as an average of the keyword embedding vectors:

$$(7.5) \quad kw = \frac{1}{n} \sum_{i=1}^n W_{kw} k_i.$$

$W_{kw}$  is the pretrained word2vec embedding matrix. In this way, the visual information is encoded as a semantic vector, and will be decoded into a piece of poetry relative to the image.

### 7.3.3 Topic-Aware Poetry Generation

To generate a poem from the semantic vector, a character-level RNN generation model is exploited then. We regard the semantic vector as the topic for a poem and input it into the RNN generation model. The output for the RNN is a poem that consists of a sequence of characters. The generation model is trained by cross entropy loss:

$$(7.6) \quad L(\delta) = -\log P_{\delta}(W = w | K = kw),$$

where

$$(7.7) \quad P_{\delta}(W = w | K = kw) = \prod_{t=1}^N p(w_t | w_1, \dots, w_{t-1}, kw; \delta).$$

However, in the conventional sequence model [75], the encoded vector would be injected by the initial states in the decoder. This may induce the RNN “forget” the initial input along the sequence generation. The generated text may deviate from the encoded vector, which induces a topic-drift problem. To tackle this problem and maintain the semantic consistency, we propose a novel topic-aware model for poetry generation.

The topic-aware model is inspired by GridLSTM [37], which contains two dimension LSTMs: a temporal LSTM and a depth LSTM. In our method, we not only inject the semantic vector through the initial state, but also inject it through the dept LSTM at every step. We use these two LSTMs to make the model “remember” topics consistently and adaptively. Specifically, We use the temporal LSTM (tLSTM) to transmit the sequence of Chinese characters, while use the depth LSTM (dLSTM) to recall the semantic vector at each step. At each time step, two LSTM computations are involved, as presented in Fig. 7.4:

$$(7.8) \quad h_t = tLSTM(w_{t-1}, h_{t-1}, C_{t-1}),$$

$$(7.9) \quad h'_t = dLSTM(w_{t-1}, h_t, kw),$$

$$(7.10) \quad p(w_t | kw, w_0, \dots, w_{t-1}) = f(h'_t),$$

where  $w_{t-1}$  is the representation of the character at  $t-1$ .  $f$  is a Softmax function with parameters  $W_s$  and  $b_s$ ,  $\text{softmax}(W_s h'_t + b_s)$ .

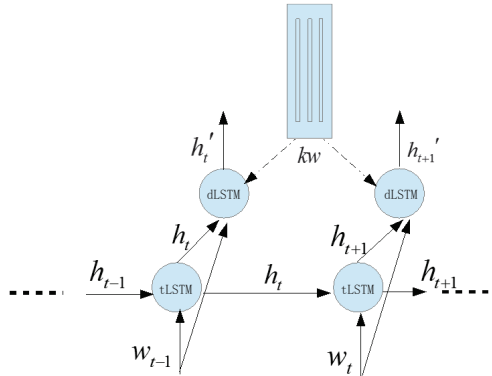


Figure 7.4: The topic-aware LSTM details.



The tLSTM works as the conventional LSTM [29] to transmit the sequence of Chinese characters:

$$\begin{aligned}
 i_t &= \sigma(U_i h_{t-1} + W_i w_{t-1} + b_i), \\
 f_t &= \sigma(U_f h_{t-1} + W_f w_{t-1} + b_f), \\
 o_t &= \sigma(U_o h_{t-1} + W_o w_{t-1} + b_o), \\
 \tilde{C}_t &= \tanh(U_C h_{t-1} + W_C w_{t-1} + b_C), \\
 C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \\
 h_t &= o_t \odot \tanh(C_t),
 \end{aligned}
 \tag{7.11}$$

where  $i_t, f_t, o_t$  are the input gate, forget gate, output gate respectively.  $C_t$  is the memory cell.  $W_{i/f/o/C}, U_{i/f/o/C}$  and  $b_{i/f/o/C}$  are parameter matrices to be learned.  $\sigma$  is the sigmoid activation function where  $\sigma(x) = 1/(1 + \exp(-x))$ .  $\odot$  denotes the product with a gate value.

The dLSTM can adaptively admit the semantic vector,  $kw$ , through the previous *memory*. Computation details in the dLSTM are as follows:

$$\begin{aligned}
 i_t' &= \sigma(U_i h_t + W_i w_{t-1} + b_i), \\
 f_t' &= \sigma(U_f h_t + W_f w_{t-1} + b_f), \\
 \tilde{C}_t' &= \tanh(U_C h_t + W_C w_{t-1} + b_C), \\
 C_t' &= f_t' \odot kw + i_t' \odot \tilde{C}_t', \\
 o_t' &= \sigma(U_o h_t + W_o w_{t-1} + b_o), \\
 h_t' &= o_t' \odot \tanh(C_t').
 \end{aligned}
 \tag{7.12}$$

The way we involve the semantic vector is novel. We not only inject it into the decoder at each step, but also inject it adaptively with the LSTM mechanism. In the dLSTM, the semantic vector,  $kw$ , is constantly recalled as the previous memory. In the depth dimension, the previous hidden state is  $h_t$ , and the input vector is  $w_{t-1}$  here. Intuitively, the input gate controls the extent to which a new value flows into the cell, and the forget gate controls the extent to which a value remains in the cell.  $kw$  interacts with the forget gate value, and new information interacts with the input gate value as shown in Equation 7.12-(4). At each step, the memory cell  $C_t'$  involves new information to update, and drop certain semantic information that is selected to be forgotten, which handles the poetry topic adaptively.

Our method is different from the attention mechanism in [4] which utilizes an extra neural network to model the alignment between inputs and the output. In our method,



we inject the semantic vector  $kw$  into the generator through the depth LSTM mechanism as described above. Compared to [4], our model is simple and efficient. Parameters in two LSTM dimensions are shared, so the semantic vector is involved without extra parameters. Although each topic word is considered in a unified way in the  $kw$ , these topic words are internally selected with more importance, and the semantic vector is selectively involved in each dLSTM. Thus, the semantic information is recalled efficiently at each step in our topic-aware poetry generation model.

In our method, the caption model is exploited to achieve visual consistency, while the generation model can achieve poem imagination. As the generation model is trained on the poem corpus, it faces poetry language constraints. The poetry generation model can always generate poetic style sentences which contain emotional words. Moreover, with the topic-aware mechanism, the generator can always generate emotional words related to the given topics. In this way, the generation model can generate emotional words compatibly even though only factual words are used as topics.

We note TA-Seq2Seq [94] and GridLSTM [37] have some connections with our work. The differences are as follows. In the topic aware sequence-to-sequence (TA-Seq2Seq) [94], the author proposed a joint attention [4] model to leverage topic information in dialogue generation. Differently, we focus on generating poems with visual semantic vectors. Moreover, we use a newly designed depth LSTM mechanism instead of the attention mechanism [4] to “remember” the topics. Compared with the joint attention mechanism, our network is straightforward and contains fewer parameters. GridLSTM [37] is a network where the LSTM cells can be arranged in a multidimensional grid. Although our topic aware generation model is inspired by GridLSTM [37], it is different in a large extent. In our work, we utilize the depth LSTM and design a novel data flow for our own task. Being injected through the previous memory cell, the topic vector gets involved adaptively as the memory cell will be updated considering the topic vector and the newly generated information.

#### 7.3.4 Anti-frequency Decoding

Generator trained with Equation 7.6 will assign higher probabilities to Chinese characters frequently appearing in the training corpus. In modern Chinese poetry collection, some words/characters, such like “I”, “heart”, “dream” appear in a higher frequency compared with others. We regard it as an imbalanced data distribution problem, where a dataset exhibits an unequal distribution between its classes. This problem is worth careful investigation as it often appears in real-world scenarios.

To solve this problem, we propose a novel decoding scheme based on Mutual Information (MI) [5]. The Mutual Information of two random variables is a measure of the mutual dependence between the two variables. Maximum Mutual Information (MMI) was introduced as objective functions in speech recognition [5] and dialogue generation [43]. In [43], an MMI-based objective function was explored to generate more diverse, interesting, and appropriate responses in dialogue generation. Thus, it is reasonable to design a decoding scheme with MI.

Our decoding scheme aims to constrain those high-frequency characters and consequently maintain the poetry readability. The mutual information between the sample semantic vector  $kw$  and the sample character sequence  $w$  in the poetry is:

$$\begin{aligned}
 I(W, K) &= \log \frac{P(K = kw, W = w)}{P(K = kw)P(W = w)}, \\
 (7.13) \quad &= \log \frac{P(W = w|K = kw)}{P(W = w)}, \\
 &= \log P(W = w|K = kw) - \log P(W = w).
 \end{aligned}$$

The second term makes a difference. It penalizes characters in the language model  $P(W = w)$ .

To merely inhibit the high-frequency characters, we take the frequency of Chinese characters in the training corpus into account. We define an alternative decoding score as:

$$(7.14) \quad s_t = \log p(w_t|w_1 \cdots w_{t-1}, kw) - \lambda g(\alpha, v_t) \log p(w_t|w_1 \cdots w_{t-1}),$$

$$(7.15) \quad g(\alpha, v_t) = \begin{cases} 1 & v_t \geq \alpha \\ 0 & v_t < \alpha \end{cases},$$

where  $\lambda$  is a penalty hyperparameter,  $v_t$  is the frequency of character  $w_t$  in the training corpus, and  $\alpha$  is the frequency threshold. Thus, high frequency words in language model  $P(W = w)$  will get penalties during decoding.

In our decoding scheme, only the high-frequency characters that appear more than frequency  $\alpha$  in the corpus would get penalties. Our model can constrain the high-frequency characters, and maintain the readability.

## 7.4 Experiments

We start with the collected poem dataset and then the implement details. To show the performance of our model, we present the results and analysis with both qualitative and quantitative evaluation.

### 7.4.1 Dataset

To train an image captioning model, we use the MSCOCO [9] dataset, which is provided for Microsoft COCO caption challenge. There are 82,783 images in the published training set, 40,504 in the validation set and 40,775 images in the test set without annotations. Each image in the training set and validation set is annotated with five descriptive English sentences. As [81, 99], we merge all published annotated data, and allocate 5,000 for validation and test split respectively. The rest 113,287 images are used for training.

For the poetry generation, we collected 16,015 modern Chinese poetries on the Internet from the beginning of the 20th century till now. Most of the poetries are collected from distinguished poets' anthologies, and some are crawled from poetry forums where poetry fans posted their compositions. To enlarge the data collection, we also added 1,189 poetries translated from other languages to Chinese by experts. The dictionary for poetry generation contains 6,194 Chinese characters. The length of poetries vary significantly. Thus, we cut long poetries into a few parts. For each poetry, we extract corresponding keywords as described in Sec. 7.3.2.2.

The keywords include nouns, adjectives as well as verbs. "world" (occurrences 2,050), "sky" (occurrences 2,294), "life" (occurrences 2,211) and "sunshine" (occurrences 1,880) are the most common topics in the poetry keyword collection.

### 7.4.2 Implementation Details

While training the captioning model, both RNN node size and word embedding size are set as 512. Dropout is 0.5 experimentally. Resnet-101[28] pretrained on ImageNet is used to encode each image. We do not rescale or crop the image. we encode the full image with the final convolutional layer, and then apply average pooling, which results in a 2048-d vector. We stop training the captioning model with cross-entropy loss after 30 epochs, and then we do RL-based training till 70 epochs. To construct the semantic vectors, the keyword embedding size is set as 512 experimentally. Maximal four keywords

are extracted from a poem sample. In the poetry generation model, both RNN size and character embedding size are set as 512. The generation model is obtained after 100 epochs of training. To get a variety of poetries instead of a unified one, we sample the characters given the decoding scores. Major details are given as below:

- LSTM size, word/character embedding size, keyword embedding size are all set as 512 experimentally.
- LSTM parameters and word/character embedding parameters are initiated by a uniform distribution in  $[-0.08, 0.08]$
- Batch size is set as 16, and the dropout rate is 0.5.
- Adam [38] is utilized for stochastic optimization. Learning rate is initially set as  $1 \times 10^{-4}$ .

Our implementation on a single Nvidia GTX 1080 GPU process at a speed of 0.32 second per iteration.

### 7.4.3 Evaluation Metrics

Evaluation of poems is a difficult task. As modern Chinese poetry is in free verse and the contents are dramatically diverse, it is hard to approximate the ground truth given the topic words. For instance, with “Love” and “Dream”, the generator can create variants of poems. So NLP metrics such as BLEU [63] is not suitable for evaluation here.

Following [31] and [104], we use human evaluation with three criteria, namely Semantic consistency, Readability and Poeticness/Aesthetics:

- Semantic consistency (S): whether the poetry is related/corresponding to the image in semantics.
- Readability (R): whether the poetry is fluent in a single sentence and/or coherent between lines.
- Poeticness/Aesthetics (P/A): Whether the poetry expressed poetically with respect to aesthetics and emotion.

We consider the overall quality of the image-poetry pair through the average of these three scores. We also report the percents of participants who consider the poetries written by human beings.

To conduct a human evaluation, we invited 46 participants to fill in questionnaires<sup>2</sup> through a WeChat platform. There are ten image-poem pairs on each questionnaire. In these ten image-poem pairs, we randomly selected one or two pairs from each model. Each questionnaire contains examples from all type of models. Participants were required to rate each pair on a 1-5 scale (the higher the better) with respect to three criteria, and distinguish whether the poem is written by human beings. Most of the participants are undergraduates or postgraduates, majored in computer science or Chinese language and Literature, from University of Technology Sydney, Beijing Institute of Technology or Peking University. Besides students, we also invited a few professors in computer science community and people engaged in art community. Participants are with an age range from 19 to 52. All of the participants took part in this challenge as volunteers. For the quality of evaluation, we only listed ten pairs on each questionnaire. Not all the 46 questionnaires are same. We designed different sets of questionnaires and asked participants to randomly choose one to fill. We required readers to read the poems patiently and analyse carefully. It would take some time to finish the questionnaire rather than scoring at the first sight roughly.

#### 7.4.4 Comparative Methods

We compare the proposed **CTAM** with three kinds of methods. 1) We extend two existing generation methods (Sequence model [75] and PPG [90]) into our task. 2) Poems written by Human and chatbot system XiaoIce are included as well. 3) We also evaluate CTAM without AFD. Details of the comparative methods as listed as follows:

**Sequence model:** The sequence model is built on the sequence-to-sequence framework[75], which contains an encoder and an decoder constructed by conventional LSTMs. According to captioning model [81], we modified the network to fit our task, where the semantic vector has been encoded and is injected into the first step of the RNN decoder.

**PPG:** A planning-based poetry generation approach [90] proposed for classical Chinese poems. Four keywords are sorted in an order manually, and a sub-topic keyword is assigned to each line. We discard the rhythmic restriction in the implementation.

**XiaoIce:** XiaoIce is a Microsoft chatbot system that can generate poems given an image. We randomly select a group of images including images from MSCOCO dataset and some others. For each image, we generate poems with all the models. For XiaoIce, we upload these images to their platform and collect the poems generated.

<sup>2</sup>One example can be accessed through <https://www.wjx.top/jq/20790601.aspx>

Table 7.2: Human Evaluation Results

Methods	S $\uparrow$	R $\uparrow$	P/A $\uparrow$	Average Score $\uparrow$	Written by Human (%) $\uparrow$
Sequence model [75]	2.45	2.45	2.95	2.62	39%
PPG [90]	2.64	2.33	2.52	2.50	35%
XiaoIce <sup>3</sup>	2.69	3.12	<b>3.39</b>	3.07	<b>56%</b>
Human	<b>3.46</b>	<b>3.42</b>	<b>3.40</b>	<b>3.43</b>	<b>64%</b>
CTAM w/o AFD	2.81	2.90	3.01	2.91	35%
CTAM (ours)	<b>2.93</b>	<b>3.38</b>	3.24	<b>3.18</b>	49%

**Human:** Humanly written poems. Obscure poems written by poets are randomly selected from our self-collected dataset. A relative image is manually matched with the poetry according to the content.

**CTAM w/o AFD:** A variation of our proposed CTAM, which performs topic-aware poetry generation without anti-frequency decoding.

## 7.4.5 Quantitative Evaluation

### 7.4.5.1 Human Evaluation

The evaluation results are reported in Tab. 7.2. Obviously, poems written by human beings obtain the highest scores across all evaluation criteria. Among the automatically generated poems, poems created by our proposed CTAM obtain the highest score, 3.18, on average. It also can be seen that our approach has an excellent performance in semantic consistency and readability. Specifically, the CTAM w/o AFD and CTAM both outperform the sequence model on semantic consistency. It can be seen that the topic-aware design is able to prevent generation deviating from the semantic vector effectively. On this aspect, our model significantly outperforms the XiaoIce, which is weak on semantic consistency. On readability, our proposed CTAM outperforms the CTAM w/o AFD. We infer that the anti-frequency decoding effectively inhibits the high-frequency words and thus the poems read more fluently. On the Poeticness/Aesthetics, XiaoIce comes second. We noted that most of the poems generated by our method are short (around four lines), while poems generated by XiaoIce mostly contain eight lines (two chunks of four lines and a blank line in the middle). By interviewing some participants, the major reason they gave XiaoIce high scores on the Poeticness/Aesthetics is that poems generated by XiaoIce have a longer form. This might also be the reason why more participants regard poems from XiaoIce as human beings written. 49% participants regard poems generated by our approach as human written. Obtaining 3.18 on average score, our CTAM outperforms all other automatic generators.

### 7.4.5.2 Ablation Study with BLEU-1

Since the proposed CTAM contains several key components, we compare variants of CTAM to demonstrate the effect of CTAM – (1) the effect of the topic-aware generator, (2) the effect of the Anti-frequency Decoding strategy. (3) We also investigate the impact of the TextRank keyword extractor. The following CTAM variants are designed:

- CTAM-d: A variant of CTAM with the dLSTM being removed. This model is the same as the LSTM-based sequence model.
- CTAM-a: A variant of CTAM with the Anti-frequency Decoding being removed.
- CTAM-k: A variant of CTAM constructed on another keyword-poem dataset in which the keywords are extracted via the word frequency.

We use BLEU-1 [63] to evaluate the generation quality of our proposed model and the model variants. BLEU-1 analyzes the co-occurrences of 1-grams between the candidate and reference poem. In the 16,015 poems, we use 10% for testing, 10% for validation and 80% for training. The results are shown in Table 7.3.

(1) *The effect of the topic-aware generator*: To verify the effect of the GridLSTM-based generator, we compare the CTAM and CTAM-d. From Table 7.3, we can easily observe that the CTAM significantly outperforms the CTAM-d on BLEU-1.

(2) *The impact of the Anti-frequency Decoding*: To see the impact of the proposed decoding strategy, we compare CTAM and CTAM-a. We note that the CTAM-a outperforms CTAM on BLEU-1, but the generations from CTAM-a lack diversity and contain a number of repeated high frequency words. For instance, there may be three fragments like “a man” in one poem. The CTAM-a gets higher BLUE-1 because these high frequency words generally appear in the reference text. The human evaluation in Table 7.2 indicates that the CTAM can generate better poems than the CTAM-a.

(3) *The impact of the keyword extractor*: To see the impact of the TextRank keyword extractor, we compare the CTAM and CTAM-k. The generation quality of CTAM is slightly better than the CTAM-k. Even though the generation quality is comparable, we need consider the keyword interaction with the caption dataset. Among these keywords, there are only 2,164 words in the caption dataset, which is less than the 2,267 intersections with the TextRank-based dataset. Thus, the TextRank algorithm is more suitable for our task.



Table 7.3: Ablation study with BLEU-1 (%)

Model	CTAM	CTAM-a	CTAM-d	CTAM-k
BLEU-1↑	9.89	12.20	5.87	9.52

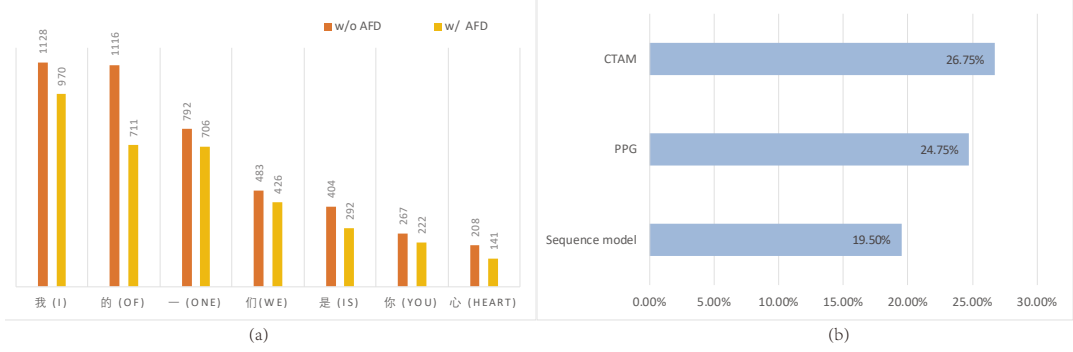


Figure 7.5: Comparison results. (a) The occurrences of seven frequently appearing characters generated by the model w/ and w/o anti-frequency decoding. (b) The percentage of generated poems containing keywords.

#### 7.4.5.3 Comparison Between Model w/ and w/o AFD

In this experiment, we compare the performance between model w/ and w/o anti-frequency decoding. We use keywords in the poetry keyword set to construct 400 keyword combinations, each of which contains maximal four keywords. A poem is generated with respect to each combination. Then, we calculate and compare the occurrences of the most commonly appearing characters.

In Fig. 7.5(a), we present the seven most commonly appearing Chinese characters and their occurrences in the generation results. We find that the occurrence of these characters from model w/AFD is clearly lower than that from model w/o AFD. The experimental results indicate that the designed decoding scheme is effective to inhibit characters with high frequency.

#### 7.4.5.4 Percentage of Generated Poems Containing Keywords

To see how the topic-aware model makes differences from others, we present the percentages of generated poems containing keywords. It is assumed that the more keywords appearing in generated poems, the more semantic consistent. To be specific, we randomly selected 400 keyword combinations to generate poems. Then, we compare the number of poems that contain the input keywords directly. Although some generation may contain words closely relative to the keywords, the comparison can somehow reveal the direct connection between keywords and poems.



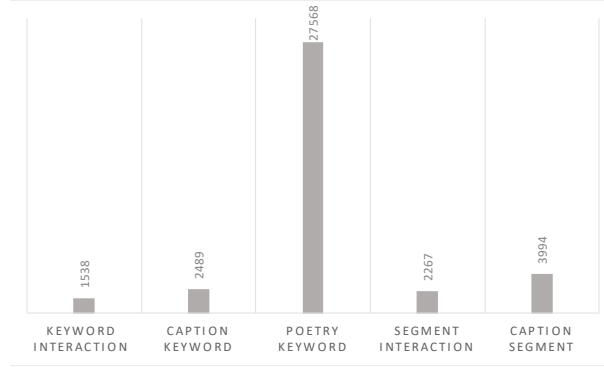


Figure 7.6: The number of poetry keywords and caption words.

As shown in Fig. 7.5(b), 26.75% poems generated by the CTAM contain keywords, while only 19.50% poems generated by the conventional sequence model contain keywords. It demonstrates that the proposed topic attention is more effective than the sequence model to maintain semantic consistency. 24.75% poems from PPG contain keywords directly, which is slightly lower than the CTAM. We note that poems from PPG contain more direct keywords than those from sequence model, but the sentences are influent sometimes. Aside from this, we find that there is a large semantic leap between two lines in PPG poems sometimes, and that the sentences are generally rigid. We infer that this may be caused by the fixed topic for each line.

#### 7.4.5.5 Comparison Between Poetry Keywords and Caption Words

To justify that the retrieval scheme is reasonable, we conduct a comparison between the number of poetry keywords and caption words. We randomly select 5,000 translated captions and do word segmentation. Then we calculate the number of caption word segments (eliminating stop words) and the number of those exist in the poetry keyword dataset. The comparison can be seen in Fig. 7.6. There are 27,568 different words in the poetry keywords dataset. 2,267 different caption words can be found in the poetry keyword dataset, which is 56.8% of 3,994 caption words. We also conduct TextRank algorithm on these 5,000 captions and extract maximal four caption-context keywords per caption. We find that 61.8% (1,538 of 2,489 caption keywords) caption-context keywords can be found in the poetry keyword dataset. In our method, maximal four visual keywords are required. Even though all the segmented words cannot be found in the poetry keyword dataset, we find the closest neighbour according to word2vec distance.

### 7.4.5.6 Image Captioning Performance

As image captioning is an important step in our approach, we present the performance for captioning as well.

Our captioning model is a variation of NIC [81]. We implement it with extra training tricks and discard the fine-tuning process. We show the performance of the model trained by cross-entropy loss and the model trained with RL algorithm respectively. Performance of a bunch of existing captioning methods are listed as well. Hard-Attention [95] is the model using attention mechanism. Review network [99] is the one involving a review network between the encoder and the decoder. ATT [102] uses image attributes as well as CNN features to generate captions.

The captioning performance is presented in Tab. 7.4. Based on [81], captioning results are evaluated by four metrics: BLEU[63], METEOR[42], CIDEr[80] and ROUGE-L[46]. BLEU analyzes the co-occurrences of n-grams between the candidate and reference sentences. METEOR is calculated by generating a 1:1 alignment between the words in the candidate and reference sentence, which is highly correlated with human judgments in settings with a low number of references. ROUGE is designed for text summarization evaluation. The CIDEr metric performs a Term Frequency Inverse Document Frequency (TF-IDF) weighting for each n-gram. As it measures consensus in image captions, CIDEr is used in the validation process to select the best model, and it is used as a reward function in the RL training.

Table 7.4: Image captioning performance

Model	BLEU-4	METEOR	ROUGE-L	CIDEr
Hard-Attention [95]	25.0	23.04	-	-
Review [99]	29.0	23.7	-	88.6
ATT [102]	30.4	24.3	-	-
CNN-RNN	31.9	25.3	53.55	99.1
CNN-RNN-RL	33.0	25.0	54.5	103.0

### 7.4.6 Qualitative Evaluation

In Fig. 7.7, we present images with captions and related keywords. It can be seen that the selected keywords contain rich visual information and can cover the major parts of the images. Thus, we believe that our visual semantic vector is useful to convey visual information, and that it's effective for semantic consistency between images and poetries.

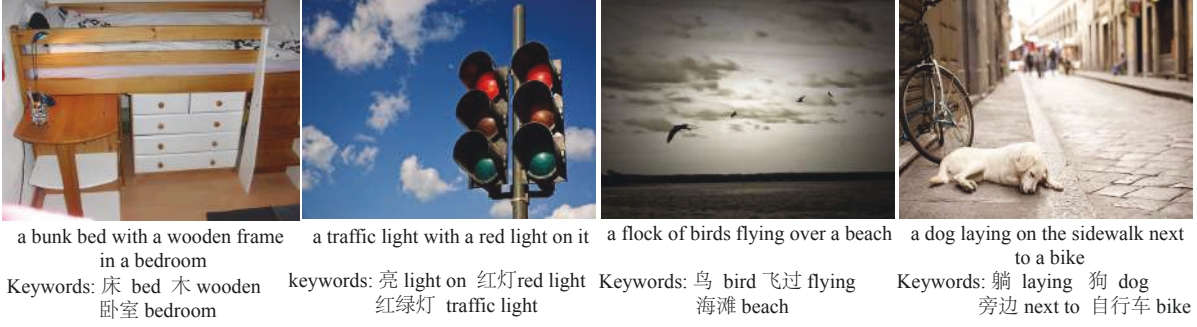


Figure 7.7: Visualization of image captions and related keywords.

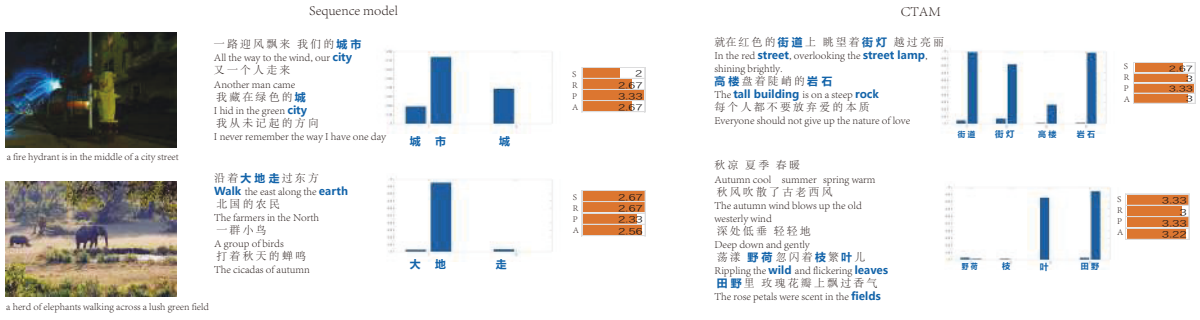


Figure 7.8: Visualization of generated results of Sequence model and the CTAM. We manually highlight relative words in poems, and present their generation probabilities in blue bars. Human evaluation results on Semantic consistency (S), Readability (R), Poeticness/Aesthetics (P) and the Average score (A) are shown in orange stripes.

A few generation examples are shown in Fig. 7.8. We present the comparison between the sequence model and the CTAM. We manually highlight the generated image-related words in the poems, and we present their probabilities in blue bars. It can be seen that the CTAM generations contain more related words compared with sequence model generations. The human evaluations are illustrated as well. For the presented poems, the CTAM outperforms the sequence model across all the criteria. The human evaluation complementarily demonstrates the effect of our proposed model.

#### 7.4.7 Effects of Hyperparameter $\lambda$

We test the effects of hyperparameter  $\lambda$  in Equation 7.14. We constrain the top 100 characters in the training corpus, and vary  $\lambda$  from 0 to 1. We select 200 keywords combination and test the frequency of character "T" among the whole generation data. The experiment results in Fig. 7.9 show that the frequency decreases accordingly with

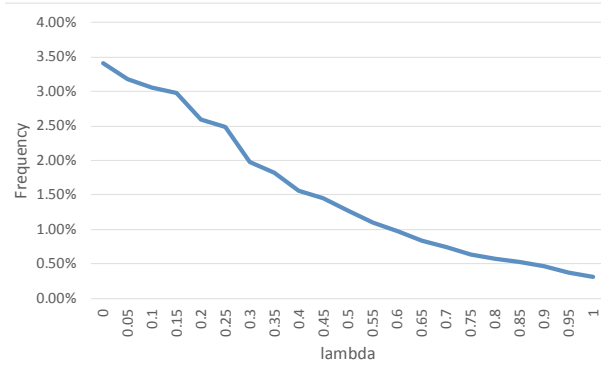


Figure 7.9: Frequency of character "I" with respect to variant hyperparameter  $\lambda$ .

the increase of  $\lambda$ . Interestingly, "I" is still among the top frequency characters in the generation, but the occurrence decreases.

## 7.5 Summary

In this chapter, we have proposed a Constrained Topic-Aware Model (CTAM) to automatically create modern Chinese poems from images. We construct a topic semantic vector incorporating with an image captioning model, which can ensure the semantic consistency between poems and images. We have proposed a topic-aware poetry generation model to avoid the topic drift problem and designed a decoding scheme to constrain high-frequency characters. Experiment results demonstrate that our approach can achieve promising performance, especially on readability and semantic consistency with the images. In the future, we plan to extend the innovative generator into other natural language processing tasks such as dialogue generation.

## CONCLUSION

## 8.1 Summary

In this thesis, we consider the problem of image captioning, aiming to develop effective methods to generate accurate and descriptive captions. To start, we reviewed existing captioning methods and introduced the basic CNN-RNN framework which is trained in two scenarios. Then, we formally presented our proposed caption generation methods from three aspects: the language model, the vision model and the training strategy. Finally, we extended image captioning to image-to-poetry generation and presented novel approaches to generate Modern Chinese Poetry from images.

Concretely, in Chapter 4, we proposed a Recall Network in the language model for the image-consistent expression. This method shows promise for preventing the language model from deviating from the image content, as the image is continually involved in the decoder. Moreover, the visual information is adaptively admitted through the inherent structure of the depth LSTM with no extra neural nets. Our proposed Recall Network can be extended and utilized in any RNN sequence generation model.

In Chapter 5, we designed new visual graphs and utilized GCN in the vision model for a comprehensive visual representation. A grid level graph is introduced to work collaboratively with the region level graph for each image. The region graph only focus on salient regions detected by detection algorithm, while the grid graph can encode fine-grained background context ignored by regional objects. With our designed edges and labels, these graph representations can incorporate reasoning ability over the visual

context and result in a comprehensive visual representation.

In Chapter 6, we proposed a novel framework and training strategy for accurate&diverse captions. We proposed an additive noise module which can manipulate the transition hidden states in the RNN decoder. We utilize RL technics to train the module, where we regard the noise module as an agent with a stochastic gaussian policy and regard generating noise as the action. During training, noiseless results are set as the baselines in the REINFORCE algorithm. This proposed method can involve adaptive perturbation in the hidden space, and thus induce diversity in the word distribution. The perturbation can somehow make the optimization algorithm jump out the local minimization. Our design is the first to pursue caption diversity&accuracy with the trainable noise module.

Finally, in Chapter 7, we extended the image captioning into an applicable task: creating Modern Chinese Poetry from images. In this task, we identified three challenges which are (a) semantic inconsistency between images and poems, (b) topic drift in the subsequent lines, (c) frequent occurrence of certain words. Regarding to the three challenges, we presented an image-to-poetry framework (Constrained Topic-Aware Model) containing three modules.

## 8.2 Future Work

Despite the rapid progress in image captioning, it is far from perfect to achieve the long-term goal: “ultimate” visual understanding. There still remains a number of challenges. In this section, I briefly list some challenges and suggestions for the future work.

**Captioning with fine-grained context.** Currently, our model only generates general factual descriptions. To understand the real world image thoroughly, more entity details are required. An example is shown in Fig. 8.1. (This image is from RichCaptioning[78]). In this example, the system can recognize the landmark as well as the celebrity. In our current model, this image can only be described as “a group of people standing on the ground”. It is obvious that these specific information should be available to the computer in supervised learning so that the generated captions may contain these visual concepts. This is related to a practical problem that is real-world data collection. This can be investigated in the future work.

**Captioning with sentiments.** In this thesis, we deal with the factual captions. In future work, captioning with sentiments can be investigated. Analysing multimedia images and generating captions with sentiments can be applied to depression detection. It would be helpful to recognize those images with strong depression and suicide tendency,



*Sasha Obama, Malia Obama, Michelle Obama, Peng Liyuan et al.  
posing for a picture with Forbidden City in the background*

Figure 8.1: An example with detailed entities. This figure is from RichCaptioning[78]

and then rescuable activity can be executed.

**Applying generators on devices.** Beyond captioning algorithms, we need to apply these algorithms on practical devices. Then, we need to consider model speeding up. Besides, edge computing can be considered when applying the generator on devices.

This thesis facilitates the connection between computer vision and natural language, and it extends the generation into specific domains. Our work makes some progress towards accurate&descriptive caption generation, but it is still an open problems. The interesting problem can still be investigated in the future work.





## REFERENCE

- [1] P. ANDERSON, B. FERNANDO, M. JOHNSON, AND S. GOULD, *Spice: Semantic propositional image caption evaluation*, in European Conference on Computer Vision, Springer, 2016, pp. 382–398.
- [2] P. ANDERSON, X. HE, C. BUEHLER, D. TENNEY, M. JOHNSON, S. GOULD, AND L. ZHANG, *Bottom-up and top-down attention for image captioning and visual question answering*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6077–6086.
- [3] L. ANNE HENDRICKS, S. VENUGOPALAN, M. ROHRBACH, R. MOONEY, K. SAENKO, AND T. DARRELL, *Deep compositional captioning: Describing novel object categories without paired training data*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1–10.
- [4] D. BAHDANAU, K. CHO, AND Y. BENGIO, *Neural machine translation by jointly learning to align and translate*, ICLR, (2015).
- [5] L. BAHL, P. BROWN, P. DE SOUZA, AND R. MERCER, *Maximum mutual information estimation of hidden markov model parameters for speech recognition*, in Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’86., vol. 11, IEEE, 1986, pp. 49–52.
- [6] K. BARNARD, P. DUYGULU, D. FORSYTH, N. D. FREITAS, D. M. BLEI, AND M. I. JORDAN, *Matching words and pictures*, Journal of machine learning research, 3 (2003), pp. 1107–1135.
- [7] D. BOSCAINI, J. MASCI, E. RODOLÀ, AND M. BRONSTEIN, *Learning shape correspondence with anisotropic convolutional neural networks*, in Advances in Neural Information Processing Systems, 2016, pp. 3189–3197.
- [8] L. CHEN, H. ZHANG, J. XIAO, L. NIE, J. SHAO, W. LIU, AND T.-S. CHUA, *Sca-cnn: Spatial and channel-wise attention in convolutional networks for image*

## REFERENCE

---

- captioning*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5659–5667.
- [9] X. CHEN, H. FANG, T.-Y. LIN, R. VEDANTAM, S. GUPTA, P. DOLLÁR, AND C. L. ZITNICK, *Microsoft coco captions: Data collection and evaluation server*, arXiv preprint arXiv:1504.00325, (2015).
- [10] X. CHEN, L.-J. LI, L. FEI-FEI, AND A. GUPTA, *Iterative visual reasoning beyond convolutions*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7239–7248.
- [11] K. CHO, *Noisy parallel approximate decoding for conditional recurrent language model*, arXiv preprint arXiv:1605.03835, (2016).
- [12] K. CHO, B. VAN MERRIENBOER, C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK, AND Y. BENGIO, *Learning phrase representations using rnn encoder–decoder for statistical machine translation*, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.
- [13] S. COLTON, J. GOODWIN, AND T. VEALE, *Full-face poetry generation.*, in ICCV, 2012, pp. 95–102.
- [14] M. CORNIA, L. BARALDI, G. SERRA, AND R. CUCCHIARA, *Paying more attention to saliency: Image captioning with saliency and context attention*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14 (2018), p. 48.
- [15] B. DAI, S. FIDLER, R. URTASUN, AND D. LIN, *Towards diverse and natural image descriptions via a conditional gan*, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2970–2979.
- [16] M. DEFFERRARD, X. BRESSON, AND P. VANDERGHEYNST, *Convolutional neural networks on graphs with fast localized spectral filtering*, in Advances in neural information processing systems, 2016, pp. 3844–3852.
- [17] J. DONAHUE, L. ANNE HENDRICKS, S. GUADARRAMA, M. ROHRBACH, S. VENUGOPALAN, K. SAENKO, AND T. DARRELL, *Long-term recurrent convolutional networks for visual recognition and description*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625–2634.

- 
- [18] P. DUYGULU, K. BARNARD, J. F. DE FREITAS, AND D. A. FORSYTH, *Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary*, in European conference on computer vision, Springer, 2002, pp. 97–112.
- [19] S. EL HIHI AND Y. BENGIO, *Hierarchical recurrent neural networks for long-term dependencies*, in Advances in neural information processing systems, 1996, pp. 493–499.
- [20] H. FANG, S. GUPTA, F. IANDOLA, R. K. SRIVASTAVA, L. DENG, P. DOLLÁR, J. GAO, X. HE, M. MITCHELL, J. C. PLATT, ET AL., *From captions to visual concepts and back*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1473–1482.
- [21] A. FARHADI, M. HEJRATI, M. SADEGHI, P. YOUNG, C. RASHTCHIAN, J. HOCKENMAIER, AND D. FORSYTH, *Every picture tells a story: Generating sentences from images*, Computer vision–ECCV 2010, (2010), pp. 15–29.
- [22] K. FU, J. JIN, R. CUI, F. SHA, AND C. ZHANG, *Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts*, IEEE transactions on pattern analysis and machine intelligence, (2016).
- [23] C. GAN, Z. GAN, X. HE, J. GAO, AND L. DENG, *Stylenet: Generating attractive visual captions with styles*, in Proc IEEE Conf on Computer Vision and Pattern Recognition, 2017, pp. 3137–3146.
- [24] Z. GAN, C. GAN, X. HE, Y. PU, K. TRAN, J. GAO, L. CARIN, AND L. DENG, *Semantic compositional networks for visual captioning*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5630–5639.
- [25] J. GU, K. CHO, AND V. O. LI, *Trainable greedy decoding for neural machine translation*, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, Sept. 2017, Association for Computational Linguistics, pp. 1968–1978.
- [26] D. HARRIS AND S. HARRIS, *Digital design and computer architecture*, Morgan Kaufmann, 2010.

## REFERENCE

---

- [27] C. HE AND H. HU, *Image captioning with visual-semantic double attention*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 15 (2019), p. 26.
- [28] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [29] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.
- [30] M. HODOSH, P. YOUNG, AND J. HOCKENMAIER, *Framing image description as a ranking task: Data, models and evaluation metrics*, Journal of Artificial Intelligence Research, 47 (2013), pp. 853–899.
- [31] J. HOPKINS AND D. KIELA, *Automatically generating rhythmic verse with neural networks*, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2017, pp. 168–178.
- [32] X. JIA, E. GAVVES, B. FERNANDO, AND T. TUYTELAARS, *Guiding the long-short term memory model for image caption generation*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2407–2415.
- [33] W. JIANG, L. MA, Y.-G. JIANG, W. LIU, AND T. ZHANG, *Recurrent fusion network for image captioning*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 499–515.
- [34] K.-C. JIM, C. L. GILES, AND B. G. HORNE, *An analysis of noise in recurrent neural networks: convergence and generalization*, IEEE Transactions on neural networks, 7 (1996), pp. 1424–1438.
- [35] J. JOHNSON, A. KARPATHY, AND L. FEI-FEI, *Densecap: Fully convolutional localization networks for dense captioning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4565–4574.
- [36] J. JOHNSON, R. KRISHNA, M. STARK, L.-J. LI, D. SHAMMA, M. BERNSTEIN, AND L. FEI-FEI, *Image retrieval using scene graphs*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3668–3678.
- [37] N. KALCHBRENNER, I. DANIHELKA, AND A. GRAVES, *Grid long short-term memory*, ICLR, (2016).

- 
- [38] D. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, ICLR, (2015).
- [39] T. N. KIPF AND M. WELLING, *Semi-supervised classification with graph convolutional networks*, ICLR, (2017).
- [40] R. KRISHNA, Y. ZHU, O. GROTH, J. JOHNSON, K. HATA, J. KRAVITZ, S. CHEN, Y. KALANTIDIS, L.-J. LI, D. A. SHAMMA, ET AL., *Visual genome: Connecting language and vision using crowdsourced dense image annotations*, International Journal of Computer Vision, 123 (2017), pp. 32–73.
- [41] G. KULKARNI, V. PREMRAJ, S. DHAR, S. LI, Y. CHOI, A. C. BERG, AND T. L. BERG, *Baby talk: Understanding and generating image descriptions*, in Proceedings of the 24th CVPR, Citeseer, 2011.
- [42] M. D. A. LAVIE, *Meteor universal: Language specific translation evaluation for any target language*, ACL 2014, (2014), p. 376.
- [43] J. LI, M. GALLEY, C. BROCKETT, J. GAO, AND B. DOLAN, *A diversity-promoting objective function for neural conversation models*, in Proceedings of NAACL-HLT, 2016, pp. 110–119.
- [44] S. LI, G. KULKARNI, T. L. BERG, A. C. BERG, AND Y. CHOI, *Composing simple image descriptions using web-scale n-grams*, in Proceedings of the Fifteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, 2011, pp. 220–228.
- [45] X. LI AND S. JIANG, *Know more say less: Image captioning based on scene graphs*, IEEE Transactions on Multimedia, (2019).
- [46] C.-Y. LIN, *Rouge: A package for automatic evaluation of summaries*, in Text summarization branches out: Proceedings of the ACL-04 workshop, vol. 8, Barcelona, Spain, 2004.
- [47] B. LIU, J. FU, M. P. KATO, AND M. YOSHIKAWA, *Beyond narrative description: generating poetry from images by multi-adversarial training*, in 2018 ACM Multimedia Conference on Multimedia Conference, ACM, 2018, pp. 783–791.
- [48] S. LIU, Z. ZHU, N. YE, S. GUADARRAMA, AND K. MURPHY, *Improved image captioning via policy gradient optimization of spider*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 873–881.

## REFERENCE

---

- [49] X. LIU, H. LI, J. SHAO, D. CHEN, AND X. WANG, *Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 338–354.
- [50] Y. LIU, J. FU, T. MEI, AND C. W. CHEN, *Let your photos talk: Generating narrative paragraph for photo stream via bidirectional attention recurrent neural networks.*, in AAAI, 2017, pp. 1445–1452.
- [51] J. LU, C. XIONG, D. PARIKH, AND R. SOCHER, *Knowing when to look: Adaptive attention via a visual sentinel for image captioning*, arXiv preprint arXiv:1612.01887, (2016).
- [52] J. MAO, X. WEI, Y. YANG, J. WANG, Z. HUANG, AND A. L. YUILLE, *Learning like a child: Fast novel visual concept learning from sentence descriptions of images*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2533–2541.
- [53] D. MARCHEGGIANI AND I. TITOV, *Encoding sentences with graph convolutional networks for semantic role labeling*, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1506–1515.
- [54] A. P. MATHEWS, L. XIE, AND X. HE, *Senticap: Generating image descriptions with sentiments.*, in AAAI, 2016, pp. 3574–3580.
- [55] R. MIHALCEA AND P. TARAU, *Textrank: Bringing order into text*, in Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.
- [56] T. MIKOLOV, K. CHEN, G. CORRADO, AND J. DEAN, *Efficient estimation of word representations in vector space*, ICLR, (2013).
- [57] V. MNIH, K. KAVUKCUOGLU, D. SILVER, A. A. RUSU, J. VENESS, M. G. BELLE-MARE, A. GRAVES, M. RIEDMILLER, A. K. FIDJELAND, G. OSTROVSKI, ET AL., *Human-level control through deep reinforcement learning*, Nature, 518 (2015), p. 529.
- [58] F. MONTI, D. BOSCAINI, J. MASCI, E. RODOLA, J. SVOBODA, AND M. M. BRONSTEIN, *Geometric deep learning on graphs and manifolds using mixture model cnns*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5115–5124.



- 
- [59] Y. NETZER, D. GABAY, Y. GOLDBERG, AND M. ELHADAD, *Gaiku: Generating haiku with word associations norms*, in Proceedings of the Workshop on Computational Approaches to Linguistic Creativity, Association for Computational Linguistics, 2009, pp. 32–39.
- [60] W. NORCLIFFE-BROWN, S. VAFEIAS, AND S. PARISOT, *Learning conditioned graph structures for interpretable visual question answering*, in Advances in Neural Information Processing Systems, 2018, pp. 8344–8353.
- [61] H. G. OLIVEIRA, *Poetryme: a versatile platform for poetry generation*, Computational Creativity, Concept Invention, and General Intelligence, 1 (2012), p. 21.
- [62] V. ORDONEZ, G. KULKARNI, AND T. L. BERG, *Im2text: Describing images using 1 million captioned photographs*, in Advances in Neural Information Processing Systems, 2011, pp. 1143–1151.
- [63] K. PAPINENI, S. ROUKOS, T. WARD, AND W.-J. ZHU, *Bleu: a method for automatic evaluation of machine translation*, in Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002, pp. 311–318.
- [64] M. PLAPPERT, R. HOUTHOOFT, P. DHARIWAL, S. SIDOR, R. Y. CHEN, X. CHEN, T. ASFOUR, P. ABBEEL, AND M. ANDRYCHOWICZ, *Parameter space noise for exploration*, ICLR, (2018).
- [65] M. RANZATO, S. CHOPRA, M. AULI, AND W. ZAREMBA, *Sequence level training with recurrent neural networks*, ICLR, (2016).
- [66] S. REN, K. HE, R. GIRSHICK, AND J. SUN, *Faster r-cnn: Towards real-time object detection with region proposal networks*, in Advances in neural information processing systems, 2015, pp. 91–99.
- [67] Z. REN, X. WANG, N. ZHANG, X. LV, AND L.-J. LI, *Deep reinforcement learning-based image captioning with embedding reward*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 290–298.
- [68] S. J. RENNIE, E. MARCHERET, Y. MROUEH, J. ROSS, AND V. GOEL, *Self-critical sequence training for image captioning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7008–7024.

## REFERENCE

---

- [69] O. RUSSAKOVSKY, J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATY, A. KHOSLA, M. BERNSTEIN, A. C. BERG, AND L. FEI-FEI, *ImageNet Large Scale Visual Recognition Challenge*, International Journal of Computer Vision (IJCV), 115 (2015), pp. 211–252.
- [70] F. SCARSELLI, M. GORI, A. C. TSOI, M. HAGENBUCHNER, AND G. MONFARDINI, *The graph neural network model*, IEEE Transactions on Neural Networks, 20 (2009), pp. 61–80.
- [71] R. SHETTY, M. ROHRBACH, L. ANNE HENDRICKS, M. FRITZ, AND B. SCHIELE, *Speaking the same language: Matching machine to human captions by adversarial training*, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4135–4144.
- [72] D. SILVER, A. HUANG, C. J. MADDISON, A. GUEZ, L. SIFRE, G. VAN DEN DRIESCHE, J. SCHRITTWIESER, I. ANTONOGLU, V. PANNEERSHELVAM, M. LANCOT, ET AL., *Mastering the game of go with deep neural networks and tree search*, nature, 529 (2016), p. 484.
- [73] K. SIMONYAN AND A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, ICLR, (2015).
- [74] B. L. STURM, J. F. SANTOS, O. BEN-TAL, AND I. KORSHUNOVA, *Music transcription modelling and composition using deep learning*, arXiv preprint arXiv:1604.08723, (2016).
- [75] I. SUTSKEVER, O. VINYALS, AND Q. V. LE, *Sequence to sequence learning with neural networks*, in Advances in neural information processing systems, 2014, pp. 3104–3112.
- [76] R. S. SUTTON AND A. G. BARTO, *Reinforcement learning: An introduction*, MIT press, 2018.
- [77] N. TOSA, H. OBARA, AND M. MINOH, *Hitch haiku: An interactive supporting system for composing haiku poem*, in International Conference on Entertainment Computing, Springer, 2008, pp. 209–216.
- [78] K. TRAN, X. HE, L. ZHANG, J. SUN, C. CARAPCEA, C. THRASHER, C. BUEHLER, AND C. SIENKIEWICZ, *Rich image captioning in the wild*, in Proceedings of



- the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 49–56.
- [79] Y. USHIKU, M. YAMAGUCHI, Y. MUKUTA, AND T. HARADA, *Common subspace for model and similarity: Phrase learning for caption generation from images*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2668–2676.
- [80] R. VEDANTAM, C. LAWRENCE ZITNICK, AND D. PARIKH, *Cider: Consensus-based image description evaluation*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4566–4575.
- [81] O. VINYALS, A. TOSHEV, S. BENGIO, AND D. ERHAN, *Show and tell: A neural image caption generator*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3156–3164.
- [82] —, *Show and tell: Lessons learned from the 2015 mscoco image captioning challenge*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (2017), pp. 652–663.
- [83] A. WANG, H. HU, AND L. YANG, *Image captioning with affective guiding and selective attention*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14 (2018), p. 73.
- [84] C. WANG, H. YANG, C. BARTZ, AND C. MEINEL, *Image captioning with deep bidirectional lstms*, in Proceedings of the 2016 ACM on Multimedia Conference, ACM, 2016, pp. 988–997.
- [85] C. WANG, H. YANG, AND C. MEINEL, *Image captioning with deep bidirectional lstms and multi-task learning*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14 (2018), p. 40.
- [86] J. WANG, J. FU, J. TANG, Z. LI, AND T. MEI, *Show, reward and tell: Automatic generation of narrative paragraph from photo stream by adversarial training*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
- [87] J. WANG, Y. PAN, T. YAO, J. TANG, AND T. MEI, *Convolutional auto-encoding of sentence topics for image paragraph generation*, arXiv preprint arXiv:1908.00249, (2019).

- [88] L. WANG, A. SCHWING, AND S. LAZEBNIK, *Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space*, in Advances in Neural Information Processing Systems, 2017, pp. 5756–5766.
- [89] Q. WANG, T. LUO, D. WANG, AND C. XING, *Chinese song iambics generation with neural attention-based model*, IJCAI, (2016).
- [90] Z. WANG, W. HE, H. WU, H. WU, W. LI, H. WANG, AND E. CHEN, *Chinese poetry generation with planning based neural network*, in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1051–1060.
- [91] R. J. WILLIAMS, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, in Reinforcement Learning, Springer, 1992, pp. 5–32.
- [92] J. WU, H. HU, AND Y. WU, *Image captioning via semantic guidance attention and consensus selection strategy*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14 (2018), p. 87.
- [93] X. WU, N. TOSA, AND R. NAKATSU, *New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system*, in International Conference on Entertainment Computing, Springer, 2009, p-p. 191–196.
- [94] C. XING, W. WU, Y. WU, J. LIU, Y. HUANG, M. ZHOU, AND W.-Y. MA, *Topic aware neural response generation.*, in AAAI, vol. 17, 2017, pp. 3351–3357.
- [95] K. XU, J. BA, R. KIROS, K. CHO, A. COURVILLE, R. SALAKHUDINOV, R. ZEMEL, AND Y. BENGIO, *Show, attend and tell: Neural image caption generation with visual attention*, in International Conference on Machine Learning, 2015, p-p. 2048–2057.
- [96] L. XU, L. JIANG, C. QIN, Z. WANG, AND D. DU, *How images inspire poems: Generating classical chinese poetry from images with memory networks*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
- [97] S. YAN, Y. XIONG, AND D. LIN, *Spatial temporal graph convolutional networks for skeleton-based action recognition*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

- 
- [98] X. YANG, K. TANG, H. ZHANG, AND J. CAI, *Auto-encoding scene graphs for image captioning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10685–10694.
  - [99] Z. YANG, Y. YUAN, Y. WU, W. W. COHEN, AND R. R. SALAKHUTDINOV, *Review networks for caption generation*, in Advances in Neural Information Processing Systems, 2016, pp. 2361–2369.
  - [100] T. YAO, Y. PAN, Y. LI, AND T. MEI, *Exploring visual relationship for image captioning*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 684–699.
  - [101] T. YAO, Y. PAN, Y. LI, Z. QIU, AND T. MEI, *Boosting image captioning with attributes*, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4894–4902.
  - [102] Q. YOU, H. JIN, Z. WANG, C. FANG, AND J. LUO, *Image captioning with semantic attention*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4651–4659.
  - [103] J. ZHANG, Y. FENG, D. WANG, Y. WANG, A. ABEL, S. ZHANG, AND A. ZHANG, *Flexible and creative chinese poetry generation using neural memory*, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1364–1373.
  - [104] X. ZHANG AND M. LAPATA, *Chinese poetry generation with recurrent neural networks.*, in EMNLP, 2014, pp. 670–680.
  - [105] Z. ZHENG, W. WANG, S. QI, AND S.-C. ZHU, *Reasoning visual dialogs with structural and partial observations*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6669–6678.
  - [106] J.-Y. ZHU, T. PARK, P. ISOLA, AND A. A. EFROS, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

