Captioning Images Containing Novel Objects

By

Yufei Wang

Supervisor: Prof. Mark Johnson

A thesis submitted to Macquarie University

in partial fulfillment of the Master By Research

Department of Computing

December 2019

MACQUARIE
University
SYDNEY·AUSTRALIA

Examiner's Copy

Typeset in LaTeX 2_ε.

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

(Signed) _____ Date: _____

Yufei Wang

Acknowledgements

I would like to thank the following people, without whom I would not have been able to complete this research, and without whom I would not have made it through my masters degree! The Center for Language Technology at Macquarie University, especially my supervisor Prof. Mark Johnson whose insight and knowledge into the subject matter steered me through this research and Dr. Ian Wood who spent countless hours discussing the detailed project plan with me and improving the English write-up for me. The Machine Learning & Perception Group at Georgia Tech, especially Dr. Peter Anderson, Mr. Karan Desai and all co-authors for `nocaps` benchmark [1] paper who provided intensive support and collaboration via slack. The Language and Social Computing Team at Data61, CSIRO, especially Dr. Stephen Wan who guided me into NLP research in recent years and provided travel funds to me to attend ACL conference during my master degree. Fellow research students, Ms Paria Jamshid Lou, Mr. Julius Lu, Mr. Yao Deng and Mr.Yupeng Jiang who gave me a lot of suggestions and always motived me towards success. Staff in Department of Computing, Macquarie University, especially Ms Joanne Aboud and Ms Karen Leung who helped me on finding my office position and travels to attend conference. Finally, my family who always support my willingness to pursue my academic goal and dream, and stand by my side even when I achieved nothing and fell behind my peers.

Abstract

Image captioning is the task of describing images using natural language. Recent advances in deep neural networks have boosted the performance of image captioning systems on large benchmark datasets such as COCO [2]. However, these data-driven approaches result in low quality captions for images containing novel objects (i.e., image objects whose corresponding textual labels are not included in the parallel image-caption training data). This thesis aims to improve generated caption quality for images containing novel objects.

We notice the limitations in previous novel object captioning benchmark and systems. The contributions of this thesis are twofold. The first contribution is a new evaluation dataset `nocaps` for novel object captioning, which is intended for evaluation of image captioning models trained on COCO. The `nocaps` benchmark is sampled from Open Images Dataset [3] with more than 400 classes of objects that are rarely seen in the COCO data. The second contribution is an improved novel object captioning model `UpDown-C`, which balances generation quality between in-domain and novel object captions.

The evaluation results show that `UpDown-C` outperforms several strong baselines, including the state-of-the-art `Up-Down` model with CBS and `NBT` model, with substantial improvement over previous work and sets a new state-of-the-art on the `nocaps` benchmark.

Contents

List of Figures

List of Tables

"Begin at the beginning", the king said, gravely, "and go on till you come to the end; then stop."

Lewis Carroll, *Alice in Wonderland*

1 Introduction

Language, one of the most shining pearls in the human intelligence, is a reflection human understandings and thoughts about our physical world. Human brains can handle visual and linguistic information jointly without explicit training. Even a two-year-old baby can point to a running dog and say "Look, that's a dog!". Image captioning is such a task that simulates this vision-to-language process. The history of this task dates back to 1966 when Marvin Minsky asked one of his undergraduate students to build up an automatic system to describe scenes from a camera [4].

A modern image captioning system takes an image as input and returns a caption for that image. Figure 1.1 shows three examples of image-caption pairs. Examples like this are used to "teach" machine learning models to talk about salient objects in the images. Although recent advances in deep neural networks have achieved impressive performance

FIGURE 1.1: Image Captioning Examples From the COCO Image Caption Benchmark.

on image captioning benchmarks, according to [5], many existing image captioning models generalize poorly to images in the wild. That is, existing image captioning models learn to mention **horse** only when the image-caption training data contains visual objects and textual mentions of **horse**, which is quite different from human beings who can describe images well once being told the exact object labels. What's more, even the largest image captioning COCO benchmark provides less than 100 types of image objects. This significantly hurts the application of automatic image captioning models in real world, such as helping people with impaired vision [6].

Novel Object Captioning [7] is a special case of image captioning that explores how to deal with the visual or textual concepts that are not present in the image-caption training data. Figure 1.2 shows a comparison of *Novel Object Captioning* and standard image captioning. They share similar training data and neural networks, but they are different in the evaluation images. In this thesis, we define *Novel Objects* as image objects whose corresponding textual labels (i.e., words or phrases) are not included in the parallel image/caption training data. Image objects whose labels are included are referred to as *Seen Objects*. Standard image captioning focuses only on images containing *Seen Objects* whilst novel object captioning deals also with images containing *Novel Objects*. In Figure 1.2, *elephant* is a *novel object* because this word never appears in the image-caption parallel training dataset. However, *horse* is a *seen objects* as it can be found in the training captions. Note that we use external sources to provide information on *novel objects*, effectively able to "see" them, such as object detection systems.

There are specialised object detectors for wide varieties of objects, e.g., animals, plants,

furniture, automobiles, etc., and it is much easier to build a specialised object detector than it is to collect captions mentioning all these objects. Instead of constructing large image-caption datasets, we consider ways of using the information produced by such object detectors in the image captioning task to generalise captioning models to *Novel Objects*. This information includes image object bounding boxes (the blue box in Figure 1.2), the extracted (visually based) features from the corresponding objects (Regions Of Interest or ROI vectors) and associated text labels. This is similar to zero-shot learning where models can recognize the concepts that are not well-trained [8]. The main idea is that, as the model sequentially constructs the caption, the caption decoder decides when and which image object labels to mention given the caption generation context and extracted object information.

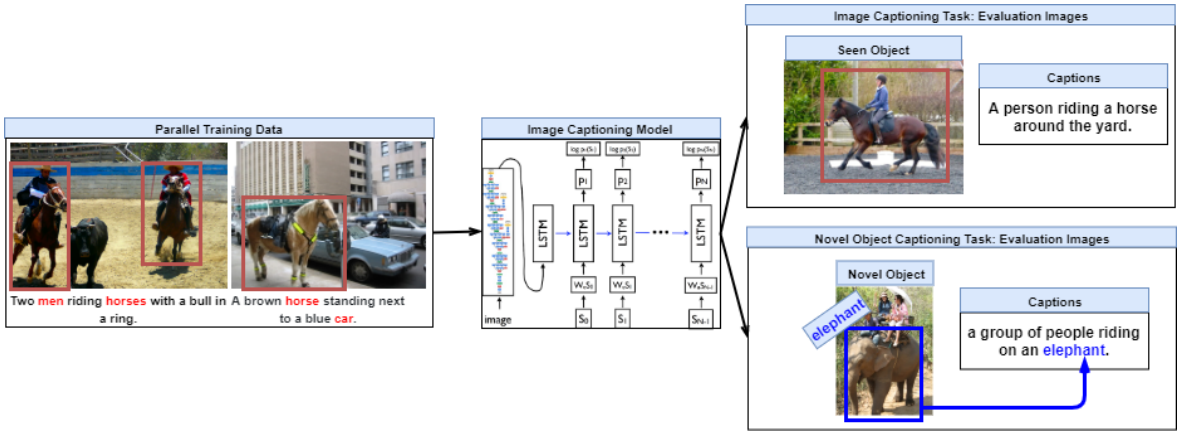

FIGURE 1.2: An comparison of Novel Object Captioning and Image Captioning.

More specifically, in this thesis, we tackle this problem in two following steps:

i) `nocaps`: A New Evaluation Dataset

Developing deep neural network models requires large and high-quality training and validation datasets. We note that the existing novel object captioning benchmark has several drawbacks and we propose an improved benchmark `nocaps` for Novel Object Captioning.

Existing approaches to novel object captioning [9–11] have been evaluated using a

proof-of-concept *COCO eight object* benchmark introduced in [7]. There are at least three limitations of this benchmark: **1)** It only has eight *novel objects* [1] held out from the COCO datase (hence, *COCO eight object*), all highly similar to existing ones, e.g. horse is a *seen object*, zebra is a *novel object*. **2)** It requires captions to train an external system that provides information about novel objects. A large number of captions are still needed in this situation. **3)** In this benchmark, *novel objects* belong to the image object label system used in the COCO training data. However, we often have to deal with image objects annotated with different scheme (i.e., Open Images labelling system).

To tackle these issues, we construct the large-scale `nocaps` benchmark from the evaluation and test split of the Open Images Dataset, which has a much larger image object label system than the COCO benchmark (600 vs. 80) and around 400 classes of objects that are rarely seen in COCO captions.

ii) `UpDown-C`: A New Captioning Model

After identifying the strengths of existing novel object captioning systems, we combine those strengths and further incorporate a novel object filtering heuristic and novel copy mechanism to produce our newly proposed `UpDown-C` model.

Copy networks [12, 13] are a special case of Seq2Seq model [14] which additionally learn to insert words from external sources (i.e.: they *copy* the words). Neural Baby Talk (`NBT`) [15] is a state-of-the-art novel object captioning system that learns to effectively talk about unseen image objects through such a copy mechanism. However, experiment results on the `nocaps` benchmark show that the state-of-the-art normal captioning model, the `Up-Down` model, in conjunction with Constrained Beam Search (CBS) [11] outperforms `NBT` by a large margin. CBS is a special inference-time decoding algorithm that enforces the use of pre-specified lexical items. CBS is disconnected from the underlying language generation model and makes sub-optimal decisions about novel object mentions. Therefore, there should be potential for captioning models with copy mechanisms to outperform models with CBS. In this thesis, we propose the `UpDown-C` model with a novel object filtering heuristic and copy mechanism which sets the new state-of-the-art on the

[1] bottle, bus, couch, microwave, pizza, racket, suitcase, and zebra

nocaps benchmark.

1.1 Contributions

In summary, this thesis makes two main contributions:

- **A large-scale Novel Objects Captioning Benchmark**. Image captioning models have achieved impressive results on datasets containing limited visual concepts and large amounts of paired image-caption training data. However, if these models are to ever function in the wild, a much larger variety of visual concepts must be learned, ideally from less supervision. To encourage the development of image captioning models that can learn visual concepts from alternative data sources, such as object detection datasets, we present the first large-scale benchmark for this task. In the computational aspect, we make a small step further: The Up-Down model can only generate words that it has seen in the image-captioning training data. To improve the performance on novel object captioning, we utilise the power of large-scale pre-trained language models so that the Up-Down model has some information about words that do not appear in the image-captioning training data. We initialise word representations and the output layer using parameters from the ELMo [16] model. Experiment results show that this improves out-of-domain performance on the nocaps benchmark.
- **New Novel Object Captioning** UpDown-C **Model**. This model builds on two state-of-the-art image captioning systems, the Up-Down and NBT models. We design a novel object filtering heuristic and copy mechanism. The UpDown-C model outperforms previous work by a large margin when decoding with beam search and sets the new state-of-the-art on the nocaps benchmark.

1.2 Thesis Outline

The remaining chapters of this thesis are organized as follows: Chapter 2 provides a general overview of the existing literature (CV and NLP) relating to novel object captioning.

Chapter 3 introduces more details about the nocaps benchmark, including data collection, characteristic and preliminary computational experiments. Chapter 4 introduces our newly proposed model UpDown-C. The UpDown-C model sets a new state-of-the-art on the nocaps benchmark. In Chapter 5, we conclude the thesis with a summary of our main contributions and a discussion of future research opportunities.

1.3 List of Publications

In this thesis, the nocaps benchmark (described in Section 3) is published as:

- H. Agrawal, K. Desai, **Y. Wang**, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee, and P. Anderson. nocaps: novel object captioning at scale. International Conference on Computer Vision (2019). https://nocaps.org

During Mres Y2 study (2019.01.15 - 2019.10.25), the author also contributed to the following projects and publications (they are not necessarily related to with thesis):

- **Y. Wang**, M. Johnson, S. Wan, Y. Sun, and W. Wang. How to best use syntax in semantic role labelling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5338-5343 (Association for Computational Linguistics, Florence, Italy, 2019). https://www.aclweb.org/anthology/P19-1529

- P. Jamshid Lou, **Y. Wang**, and M. Johnson. Neural constituency parsing of speech transcripts. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1(Long and Short Papers), pp. 2756-2765 (Association for Computational Linguistics,Minneapolis, Minnesota, 2019). https://www.aclweb.org/anthology/N19-1282

- Yifang Sun, Shifeng Liu, **Yufei Wang**, Wei Wang. Extracting Definitions and Hypernyms with a Two-Phase Framework. The 24th International Conference on Database Systems for Advanced Applications (DASFAA 2019), 2019. https://link.springer.com/chapter/10.1007/978-3-030-18590-9_57

"All of us are interested in our roots."

Donald E. Osterbrock, *Organizations and Strategies in Astronomy*

2

Background and Related Work

2.1 Computer Vision Background Knowledge

2.1.1 Image Classification

Image classification is the task of assigning a class label to input images with a probability indicating how likely the input image belongs to that class. This task can either be a single-class classification task where the sum probability of all labels equals to 1, or a multi-label classification task [17] where the output is a set of image labels (e.g., a dog and a cat). The existing state-of-the-art systems for image classification are Convolutional Neural Networks (CNN) [18] with various architectures, such as VGG [19] and ResNet [20]. Pre-trained CNN models, which are trained on large amounts of image-label pairs, are used as powerful feature extractors for many down-stream tasks, such as object detection.

2.1.2 Object Detection

Object detection is a fine-grain image understanding task that identifies bounding boxes that cover entire objects and assigns objects to pre-defined categories simultaneously. Figure 2.1 compares image classification to object detection. Modern neural object detection systems can be categorised into two-stage systems, such as Fast R-CNN [21], Faster R-CNN [22] and one-stage systems, such as SSD [23] and YOLO [24]. The two-stage object detectors usually have better performance than their one-stage counterparts, whilst one-stage object detectors usually enjoy faster speed with lower memory requirements.

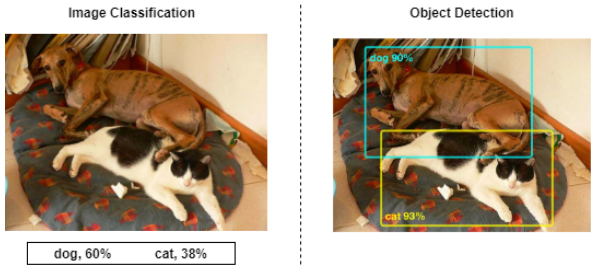

FIGURE 2.1: A comparison between image classification and object detection outputs. Both tasks detect a dog and a cat from the given image, but in different ways.

In this thesis, a two-stage Faster R-CNN object detector is deployed to identify image objects for image captioning systems. Figure 2.2 shows the standard architecture of a Faster R-CNN object detector. The first stage proposes object bounding boxes using the base CNN features and the second one computes a *Region of Interest (ROI) vector* for each proposal via *ROI Pooling* which transforms a bounding box (potentially from an external source) into a fixed-sized dense vector. The *ROI vector* is used as input to an object classifier and to refine the bounding box position via *Box Regression*. The state-of-the-art image captioning models, `Up-Down` and `NBT`, use *ROI vector* in their visual attention modules.

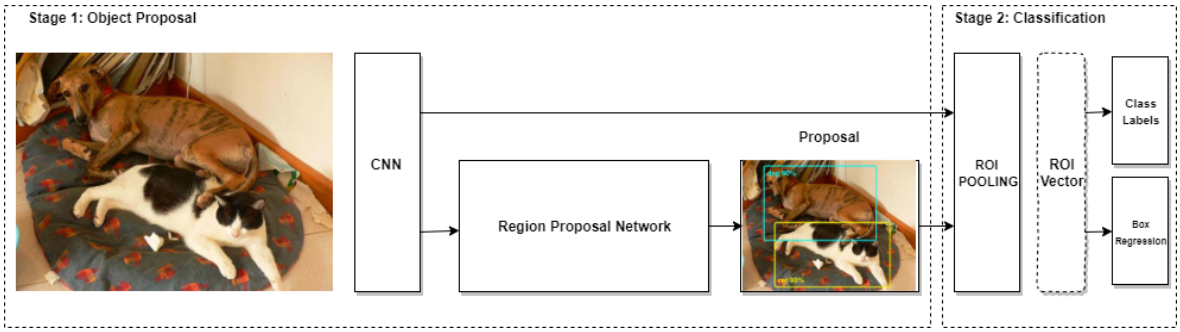

FIGURE 2.2: Faster R-CNN architecture

2.2 Natural Language Process Background Knowledge

2.2.1 Large-scale Pre-trained Language Models (LM)

The idea of learning word representations from large-scale text corpora has a long history in the NLP community [25–27]. [28] proposes *word2vec*, a reduced version of a forward-feed neural language model proposed in [29], to learn a set of dense vectors for words encoding their semantics. These fixed word vectors are widely used in neural networks, such as Recurrent Neural Networks (RNN). As computational power has increased, training large-scale context-aware pre-trained language models has become possible. The first such model was ELMo [16] which uses the linear-weighted outputs of a large bi-LSTM model to represent input words. These word representations vary given different sentence contexts. The follow-up works, Bert [30], GPT [31] and XLNet [32] train large transformers [33] to learn word representations using larger training data and novel loss functions other than the standard left-to-right language modeling loss.

As shown in Table A.1, these language models are all trained by increasingly large text corpora, which increases their representational power. When incorporating word representations from these language models in down-stream tasks, such as Question Answering [34] and Semantic Role Labelling [35], the performance is significantly improved. In this thesis, to handle words from unseen image objects, parameters from ELMo are used to initialize some important parameters of our image caption decoder.

2.2.2 Copy Networks

The original copy network, namely the pointer network [12], is adapted from a *sequence-to-sequence* model [14] to solve problems whose output size depends on their input size, such as the *Travelling Salesman Problem* and *Delaunay Triangulation*. Instead of predicting from pre-specified output schemes (e.g., fixed vocabularies), pointer networks predict the input item order (i.e., which input items come first, second etc.).

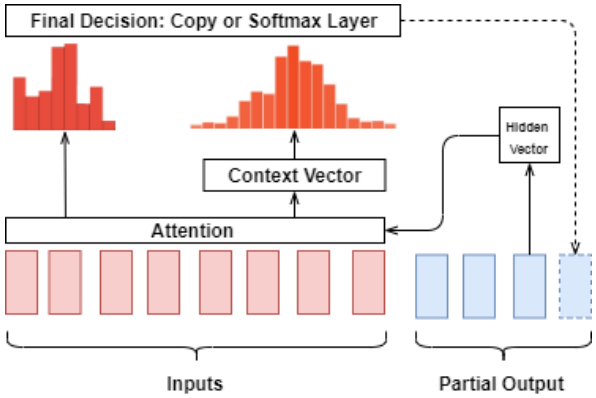

FIGURE 2.3: A General Copy Network Architecture

The follow-up works [13, 36, 37] adapt this idea to the *sequence-to-sequence* models used in machine translation and text summarisation. These copy networks either generate words from the standard output layer, or copy words from external sources, such as input sequences. The main motivation is to copy out-of-vocabulary words from the inputs. As shown in Figure 2.3, a common copy network is based on a *sequence-to-sequence* model with attention over input tokens. The input attention weights are used as copy probability which is then compared with the standard output probability in order to make the final copy decision.

2.3 Image Captioning

One important breakthrough of deep neural networks is to represent everything in the network as dense vectors. Both words and image regions are represented in the same feature space. This revolutionary change makes image captioning systems with and

without deep learning technologies different, which will be introduced separately. We also talk about the common evaluation metrics used for image captioning task.

2.3.1 Image Captioning Before Deep Learning

Text generation systems without deep learning usually use image object labels, attributes and relationships to fill pre-defined templates. [38] proposes Visual Dependency Representations (VDR), similar to dependency graphs, to capture object spatial relations and traverse them to fill the templates. [39] uses image object information to generate a syntax tree that describes the image content. [40] trains an n-gram language model to generate function words to connect key components in the scene-graphs.

Another approach is to "generate" captions by retrieving from training captions [41–43]. All output captions are actually from the training captions. Given the input images, these systems first look for similar training images and then copy from the corresponding captions. For example, [43] proposes to map sentence and image representation into the same space. A similarity score can be calculated from this space given image-caption pairs. [42] first looks for similar images and selects the most representative captions by using lexical overlap with other training captions. They show a strong baseline on COCO, the largest and most widely used benchmark for image captioning.

2.3.2 Image Captioning with Deep Learning

Figure 2.4 shows a typical neural image captioning architecture which includes a Convolutional Neural Network (CNN) as an image encoder and a Recurrent Neural Network (often, LSTM [44]) as a text decoder. [45] uses a LSTM decoder with an attention module that operates on a set of uniformly-divided feature vectors from a CNN encoder. The system is trained to focus on the salient and relevant image regions when generating captions.

Note that the feature vectors from a CNN model are usually associated with uniformly-divided grids on the input images. However, for image captioning, salient parts of the images are often much more important than other parts. To model the salience of the

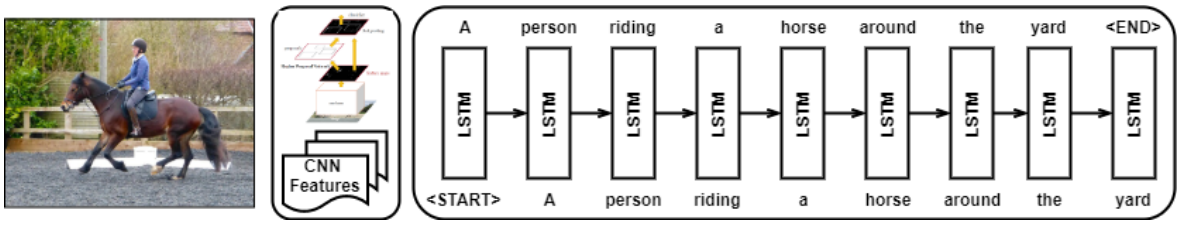

FIGURE 2.4: Basic CNN-RNN Image Captioning System

input images, [46–48] extract vectors representing high-level semantic elements, such as image objects and labels from external object detectors. The *ROI vectors* of these extracted image objects are used as the basis of the visual attention in these models.

2.3.3 Evaluation Metrics

Given an output caption y, a set of ground truth references $\mathbb{S} = \{s_1, \ldots s_k\}$ and an image I, the goal of image captioning evaluation metrics is to estimate quality of y given I. The common idea is to discard the visual information in I and only compare the linguistic similarity between y and ground truth references $\mathbb{S}$, making it a NLP analysis problem. Bleu [49] and METEOR [50] are proposed to evaluate machine translation systems by calculating n-gram precision and synonym matching between the outputs and ground truth references. ROUGE [51], a text summarization evaluation metric, is based on n-gram recall. CIDEr [52] and SPICE [53] are all specially designed for image captioning task. CIDEr focuses on measuring tf-idf weighted n-gram cosine similarity between the output captions and ground truth ones. SPICE is built on scene graph (See an example of scene graph in Figure A.1). In summary, Bleu, METEOR, ROUGE and CIDEr are evaluating the lexical overlap and SPICE measures the preserved structural semantic meaning in the output captions.

2.4 Novel Object Captioning

In *Novel Object Captioning*, "novel objects" are defined as those image objects whose corresponding textual labels (i.e., words or phrases) are not included in the parallel

image-caption training data. This presents a big challenge for captioning models to learn how to mention those unknown labels in the captions. In this section, we introduce three types of Novel Object Captioning models.

2.4.1 Decoupled Language Model for Novel Objects

[7, 54] attempt to decompose the image captioning task into visual and linguistic sub-modules that can be trained independently as well as jointly. In the independent training phrases, images and sentences that contain novel object information are used in the visual and textual sub-module separately. They are also jointly trained using the available image-caption dataset. Their experiments show that the joint models can learn to talk about novel objects fluently.

2.4.2 Copy Network for Novel Objects

As mentioned above, copy networks can generate out-of-vocabulary words from input sequences. In the context of novel object captioning, [9, 10, 15, 55] propose novel copy networks to copy from image object labels. These systems share a similar framework: a standard text decoder, a copy mechanism and a novel object information provider. The text decoders are usually a two-layer LSTM with an attention module in the middle. The copy mechanisms decide when to copy and which particular object labels to copy. Object detectors [15, 55] or image classifiers [9, 10] are often used to characterize novel objects in the input images.

For example, when generating a caption for the image with *elephant* in Figure 2.5, instead of generating all the words, copy networks would generate ***a group of people riding on an slot*** in the first step. The *slot* is tied to a particular object in the input image. It is then refined to a concrete word and caption generation continues.

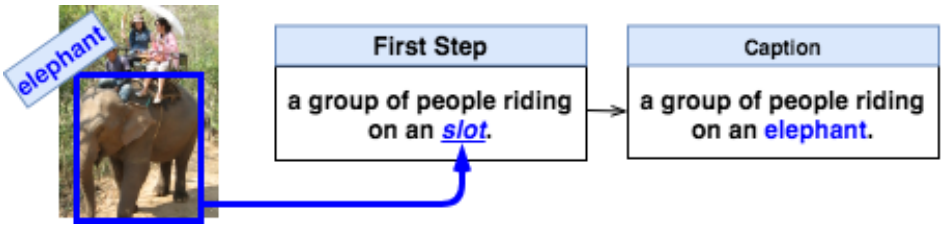

FIGURE 2.5: A Running Example of Copy Networks for Novel Object Captioning

2.4.3 Constrained Beam Search for Novel Objects

Constrained Beam Search (CBS) [11] and its follow-up studies [56, 57] propose sophisticated beam search algorithms that enforce the inclusion of pre-specified lexical items in captions. CBS with beam size K maintains a top-K beam in each state of a finite-state machine (FSM). In each generation step, each state updates its top-K beam from all other connected states. Take the image containing *elephant* in Figure 1.2 as an example. An FSM with $2^2 = 4$ states is required to maintain two constraints $D_1 = people$ and $D_2 = elephant$. As shown in Figure 2.6, q_1 keeps the top-K captions without *people* and *elephant*. Once the decoder generates captions with either *people* or *elephant*, they move to either q_2 or q_3. Finally, captions in q_4, the accepting state, satisfy both constraints *people* and *elephant*. The captions stay in the same states when mentioning none of the constraints. CBS only compares captions in the same state (i.e., satisfying the same constraints), avoiding the problem of extremely low prediction probability of unseen words. [58] apply CBS into the training stage in an EM-like iterative algorithm [59] to further improve performance on novel object captioning tasks.

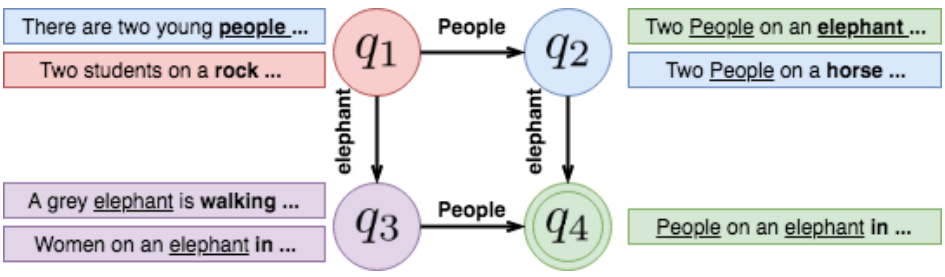

FIGURE 2.6: FSM for $D_1 = people$ and $D_2 = elephant$ constraint. This is a snapshot after generating 5^{th} word. Caption colors indicate the new states to which captions will move.

2.5 Datasets

Large-scale and high quality training datasets are critical to the success of neural image captioning systems. This section introduces important datasets relevant to this thesis, including Image Captioning benchmarks (Section 2.5.1), ImageNet (Section 2.5.2), Visual Genome (Section 2.5.3) and Open Images (Section 2.5.4) datasets.

2.5.1 Image Captioning Benchmarks

Table A.2 summarizes the meta information of popular image caption benchmarks. COCO, Flickr [1] 8K and Flickr 30K are crowd-annotated benchmarks, which are collected using crowd workers with carefully curated instructions to control the quality and style of the resulting captions. Each image is usually associated with multiple captions to improve the reliability of automatic evaluation metrics. However, they cover limited visual concepts. For example, the COCO benchmark has less than 100 classes of objects. Im2Text, Pinterest40M and Conceptual captions are automatically collected benchmarks, which are mostly sourced from web pages. These benchmarks contain many diverse visual concepts, but are also more likely to contain non-visual content in the description due to the automatic collection pipelines. These automatic benchmarks only include one caption per image and lack human baselines.

2.5.2 ImageNet

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC or ImageNet) [60] is a popular benchmark and challenge for image classification and object detection. The dataset contains photographs collected from the well-known photo and video hosting website Flickr and other search engines, manually annotated by annotators from the Amazon Mechanical Turk [2] crowd-sourcing platform. Various CNN architectures (e.g., VGG [19] and ResNet [20]) pretrained on ImageNet are used as powerful feature extractors for downstream tasks, such as object detection.

[1] https://www.flickr.com
[2] https://www.mturk.com

2.5.3 Visual Genome

The Visual Genome Dataset [61] is a collection of short free texts for object regions in the images obtained via crowd-sourcing. Figure 2.7 shows the two-stage crowd-sourcing procedure for the Visual Genome Dataset. **i)** given an input image, the annotators draw bounding boxes and write short descriptions for the corresponding regions. **ii)** Another group of annotators give tight bounding boxes, names, attributes and relationships for all objects in all regions. These annotations allow us to train object detection models. The Visual Genome Dataset provides rich contextual visual information, allowing models to learn a better alignment between visual and textual representations.

FIGURE 2.7: Visual Genome Annotation Stages.

2.5.4 Open Images

The Open Images Dataset [3] [3] is a large benchmark with multi-level annotations, including image-level labels, object bounding boxes, object segmentation masks, and visual object relationships. All images are first collected from Flickr. The simple and popular images are then filtered out so the resulting images are very diverse and often contain complex scenes with several objects (8.3 per image on average). The Open Images Dataset uses 600 image object labels which are organized in a tree structure (see Figure A.2 for details). We explore the use of this tree structure when representing novel objects.

[3] https://storage.googleapis.com/openimages/web/index.html

"When you make the decision to start something new, first figure out the jobs you want to do. Then position yourself to play where no one else is playing.!"

Whitney Johnson

3

nocaps: novel object captioning at scale

The nocaps benchmark [1] is collected from the Open Images Dataset [3], which is the largest available human-annotated object detection dataset with complex scenes annotated with object bounding boxes for 600 object classes. Moreover, out of the 500 classes that are not overly broad (e.g. 'clothing') or infrequent (e.g. 'paper cutter'), nearly 400 are never or rarely mentioned in the COCO benchmark [2] which we select as image-caption training data, making these images an ideal basis for our benchmark. Note that we need the training split of the Open Images Dataset to build an external object detector for novel object image captioning systems. So all images in the nocaps benchmark are from the

[1] Note that some materials in this chapter come from our recently published paper [1]. I am the third author of this paper (the first two authors contribute equally). In [1], my contributions are to implement the ELMo-enhanced Up-Down model and to contribute to implementation of the Neural Baby Talk (NBT) model, as well as the corresponding experimental results analysis.

validation and test split of the Open Images Dataset.

3.1 nocaps Data Collection

3.1.1 Image Selection

Since Open Images Dataset is primarily an object detection dataset, a large fraction of images contain well-framed iconic perspectives of single objects. The distribution of object classes is highly unbalanced, with a long-tail of object classes that appear relatively infrequently. For image captioning tasks, images with multiple objects and rare object co-occurrences are more interesting and challenging. This motivates the following image selection steps:

– Excluding all images with unknown rotation.
– Excluding all images with single ground truth object category.
– Including all images with more than 6 unique ground truth object categories.
– Randomly selecting remaining images in a way that improves the entropy over object classes in selected set of images.

3.1.2 Collecting Human Annotations

In the nocaps benchmark, 11 English language captions [2] for each image are collected using a large pool of crowd-workers on Amazon Mechanical Turk (AMT). Figure 3.1 shows the nocaps annotation interface. Our image caption collection interface closely resembles the interface used for collection of the COCO Captions dataset, albeit with one important difference. We prime workers by displaying the list of ground-truth object classes present in the image (as indicated by the red box). To minimize the potential for this priming to reduce the language diversity of the resulting captions, the object classes were presented as 'keywords', and workers were explicitly instructed that it was not necessary to mention all the displayed keywords. To reduce clutter, we did not display object classes belonging

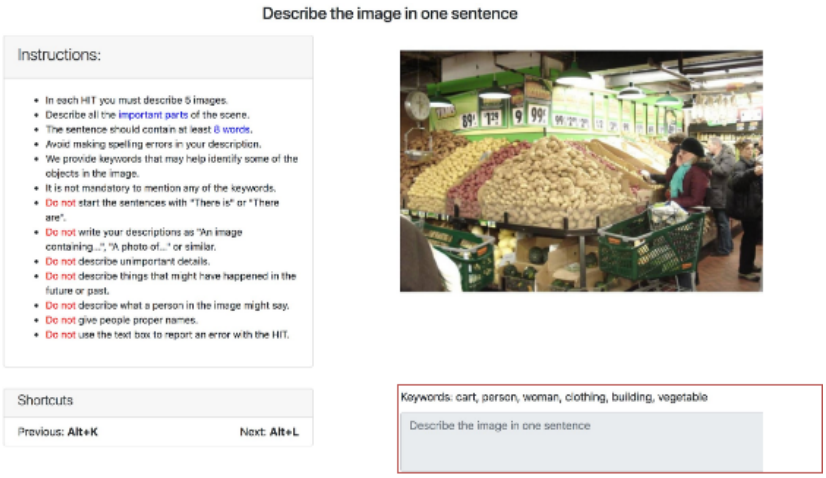

FIGURE 3.1: nocaps annotation interface. The red box indicates the hints provided to the annotators.

to *parts* of their parent classes, such as human hand, tire, door handle.

3.1.3 Comparison between nocaps and COCO

Figure 3.2 compares object categories, number of instances and word counts between the COCO and nocaps benchmarks. Since the Open Images object label system is much larger than the object label system in the COCO benchmark (600 vs. 80), nocaps contains more object classes per image (4.0 vs 2.9), and slightly more object instances per image (8.0 vs 7.4). We investigated whether priming annotators negatively reduces language diversity. We see in Table 3.1 that nocaps has a larger vocabulary (1-grams) and more diverse language compositions (2-, 3- and 4-grams) than the COCO benchmark.

Dataset	1-grams	2-grams	3-grams	4-grams
COCO	6,913	46,664	92,946	119,582
nocaps	8,291	59,714	116,765	144,577

TABLE 3.1: Unique n-grams in equally-sized (4,500 images / 22,500 captions) uniformly randomly selected subsets from the COCO and nocaps validation sets.

[2] 1 caption for human baseline and 10 captions for evaluation.

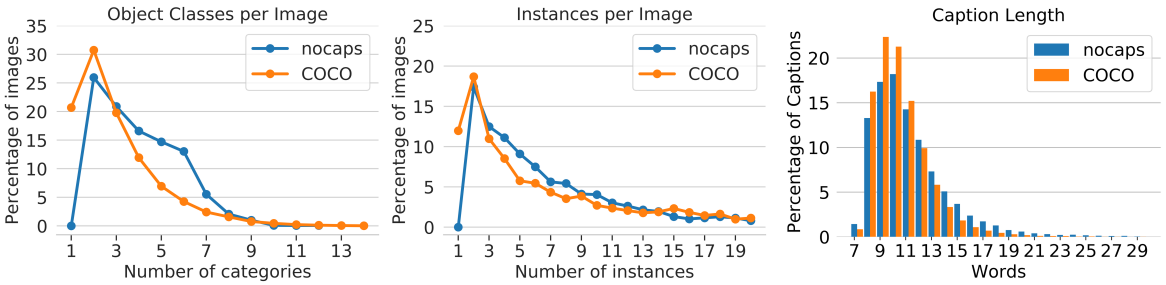

FIGURE 3.2: Compared to COCO Captions [2], on average nocaps images have more object classes per image (4.0 vs. 2.9), more object instances per image (8.0 vs. 7.4), and longer captions (11 words vs. 10 words).

3.2 nocaps Evaluation Framework

The aim of nocaps is to benchmark progress towards models that can describe images containing visually novel concepts. To avoid exposing the novel object captions, an evaluation server [3] is hosted on EvalAI [62] for fast evaluation with the following guidelines: **i) Do not use additional paired image-caption data.** Improving evaluation scores by leveraging additional paired data is antithetical to this benchmark – *the only paired image-caption dataset that should be used is the COCO 2017 training split*. However, other datasets such as external text corpora, knowledge bases, and additional object detection datasets may be used during training or inference. **ii) Do not leverage ground truth object annotations.** We note that ground-truth object detection annotations are available for the Open Images validation and test splits (and hence, for nocaps). While ground-truth object annotations may be used to establish performance upper bounds on the validation set, they should never be used in a submission to the evaluation server unless this is clearly disclosed.

The main metrics used for nocaps are CIDEr [52] (C) and SPICE [53] (S) because they are shown to have the strongest correlation with human judgments, but performance on Bleu [49] (B-1, B-4), Meteor [50] (M) and ROUGE [51] (R) will also be reported.

To demonstrate the strengths of different captioning models, we divide the images in the nocaps benchmark into three subsets: `in-domain`, `near-domain` and `out-of-domain`. Images with only non-novel objects are classified as `in-domain` and

[3] https://evalai.cloudcv.org/web/challenges/challenge-page/355/overview

images with only novel objects are classified as out-of-domain. The remaining images are classified as near-domain. The 80 COCO object categories and 39 Open Images object categories that are frequent ($\geq$ 1000) in the COCO training captions are recognized as non-novel objects and the remaining as novel objects. 87 Open Images object categories are not used in the nocaps benchmark. Table 3.2 shows the size of the three splits of the nocaps benchmark test dataset. We note that near-domain is the largest split as it is important to handle the relationships between seen objects and novel objects.

Item	in-domain	near-domain	out-of-domain
Images	1311	7406	1883
Captions	13K	74K	19K

TABLE 3.2: Size of three splits of the nocaps benchmark test dataset.

3.3 Baseline Models for nocaps

In this section, we introduce the details of two computational models tested on the nocaps benchmark, the Up-Down model and the Neural Baby Talk (NBT) model. We also discusse how we apply constrained beam search to both models.

3.3.1 Image Object Inputs

Both the Up-Down and NBT models use two sets of image objects $\mathbb{O}^{vg} = \{o_1^{vg}, o_2^{vg}, \ldots, o_{k_{vg}}^{vg}\}$ and $\mathbb{O}^{oi} = \{o_1^{oi}, o_2^{oi}, \ldots, o_{k_{oi}}^{oi}\}$. $\mathbb{O}^{vg}$ are extracted with an object detector trained on the Visual Genome Dataset [61]. They are originally proposed in [46]. $\mathbb{O}^{oi}$ are extracted with a pre-trained object detector [4] trained on the Open Images Dataset [3] from the Tensorflow model zoo [63]. They are treated as copy candidates because the nocaps benchmark annotators use Open Images ground truth image object labels in the captions. Both object detectors use the Faster-R-CNN [22] architecture.

[4] tf_faster_rcnn_inception_resnet_v4_atrous_oidv2

In this thesis, the NBT model uses both image objects $\mathbb{O}^{vg}$ and $\mathbb{O}^{oi}$ together and the Up-Down model only uses image objects $\mathbb{O}^{vg}$ because it does not learn to copy from external sources. The constraints used in Constrained Beam Search are calculated from image objects $\mathbb{O}^{oi}$.

3.3.2 ELMo-enhanced Up-Down Model

The Up-Down model follows the standard neural text generation architecture. Figure 3.3 shows an example when the Up-Down model generates caption for an input image. Each step recurrently uses the word generated in the last step. Special *<START>* and *<END>* tokens control the length of output captions. The NBT and UpDown-C models are all built on top of this generation mechanism. In this section, to handle novel object labels, we use ELMo parameters to initialize the output layer and use ELMo output to encode input words.

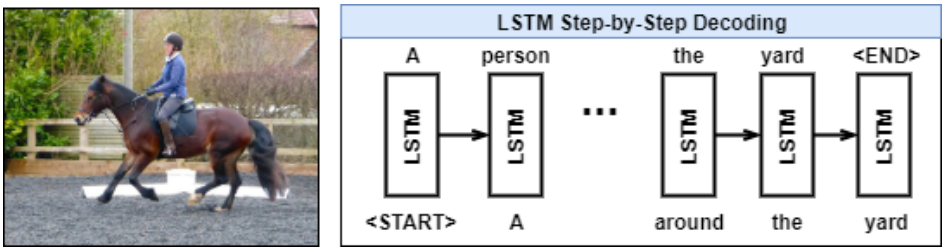

FIGURE 3.3: An example when the Up-Down model generates *A person riding a horse around the yard.*

ELMo Details

ELMo is a large-scale pretrained language model. Figure 3.4 shows the ELMo architecture on the right box. It uses a character CNN [64] to represent input words, denoted as w_{char}. We have:

$$\boldsymbol{w}_c = CNN(w_{char}) \tag{3.1}$$

The input representation w_c is then passed to a two-layer LSTM model ($LSTM_1^e$ and

$LSTM_2^e$) with a skip-connection in the last layer. We have:

$$\boldsymbol{h}_{t,1}^e = LSTM_1^e(\boldsymbol{w}_c) \tag{3.2}$$

$$\boldsymbol{h}_{t,2}^e = LSTM_2^e(\boldsymbol{h}_{t,1}^e) + \boldsymbol{h}_{t,1}^e \tag{3.3}$$

A standard left-to-right language model predicts each word given the previous words. This is given by:

$$P(y_t|y_{1:t-1}) = softmax(\boldsymbol{W}^e \boldsymbol{h}_{t,2}^e + \boldsymbol{b}^e) \tag{3.4}$$

where $\boldsymbol{W}^e \in \mathbb{R}^{H^e \times V}$ and $\boldsymbol{b}^e \in \mathbb{R}^v$. H^e is the ELMo hidden state size and V is the ELMo vocabulary size. ELMo is trained using standard cross-entropy loss.

Using ELMo LSTM output vectors (i.e., $\boldsymbol{w}_c$, $\boldsymbol{h}_{t,1}^e$ and $\boldsymbol{h}_{t,2}^e$) as word representations can generally improve the performance of downstream tasks [65, 66]. In this thesis, we use the linear-weighted sum of $\boldsymbol{w}_c$, $\boldsymbol{h}_{t,1}^e$ and $\boldsymbol{h}_{t,2}^e$ as the word representation $\boldsymbol{w}_t^e$ for image captioning models

$$\bar{\gamma_0}, \bar{\gamma_1}, \bar{\gamma_2} = softmax(\gamma_0, \gamma_1, \gamma_2) \tag{3.5}$$

$$\boldsymbol{w}_t^e = \bar{\gamma_0} \cdot \boldsymbol{w}_c + \bar{\gamma_1} \cdot \boldsymbol{h}_{t,1}^e + \bar{\gamma_2} \cdot \boldsymbol{h}_{t,2}^e \tag{3.6}$$

where $\gamma_i \in \mathbb{R}$ (i=0, 1, 2) are trainable scalars. When using $\boldsymbol{w}_t^e$ as the external word representation in the image captioning models, all the parameters of ELMo but γ_i (i=0, 1, 2) are fixed during training.

Integrating ELMo into the `Up-Down` Model

As shown in Figure 3.4, the `Up-Down` model [46] is a two-layer LSTM model ($LSTM_{vis}$ and $LSTM_{txt}$, both of dimension M) with a visual attention module between the LSTM layers.

In the `Up-Down` model, each input image is represented as a set of Visual Genome *ROI vectors*. Given by:

$$\mathbb{R}^{vg} = \{\boldsymbol{r}_1^{vg}, \boldsymbol{r}_2^{vg}, \ldots, \boldsymbol{r}_{k_{vg}}^{vg}\}, \ \boldsymbol{r}_i^{vg} \in \mathbb{R}^D \tag{3.7}$$

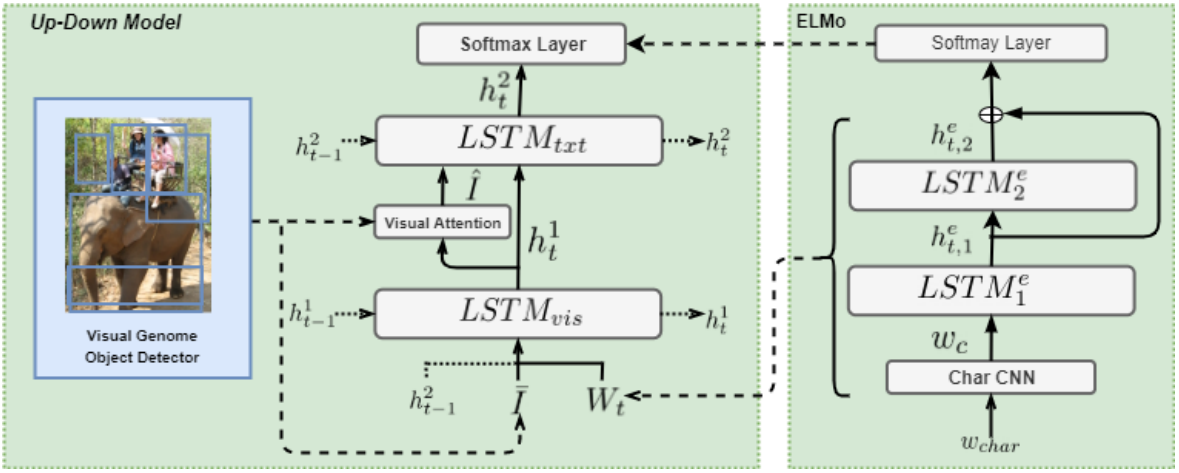

FIGURE 3.4: An overview of ELMo-enhanced Up-Down Baseline

In step t, we use the ELMo word representation $\boldsymbol{w}_t$ of the previously generated word y_{t-1} and averaged ROI features:

$$\boldsymbol{w}_t = ELMo(y_{t-1}) \tag{3.8}$$

$$\bar{I} = \frac{1}{k_{vg}} \sum_{i=1}^{k_{vg}} \boldsymbol{r}_i^{vg} \tag{3.9}$$

The input to $LSTM_{vis}$ is then the concatenation $\boldsymbol{x}_t^1 = [\bar{I}, \boldsymbol{h}_{t-1}^2, \boldsymbol{w}_t]$ where $\boldsymbol{h}_{t-1}^2 \in \mathbb{R}^M$ is the previous hidden state from $LSTM_{txt}$. Given $\boldsymbol{h}_t^1$, the output of $LSTM_{vis}$, we calculate the attention score of $\boldsymbol{h}_t^1$ over image objects $\boldsymbol{r}_i^{vg}$ as the visual input to $LSTM_{txt}$, given by:

$$a_{t,i}^{vg} = \boldsymbol{w}_{vis}^T tanh(\boldsymbol{W}_r \boldsymbol{r}_i^{vg} + \boldsymbol{W}_h \boldsymbol{h}_t^1) \tag{3.10}$$

$$\bar{\boldsymbol{a}}_t^{vg} = softmax(\boldsymbol{a}_t^{vg}) \tag{3.11}$$

$$\hat{I} = \sum_{i=1}^{k_{vg}} \bar{\boldsymbol{a}}_{t,i}^{vg} \cdot \boldsymbol{r}_i^{vg} \tag{3.12}$$

where $\boldsymbol{W}_r \in R^{H \times D}$, $\boldsymbol{W}_h \in R^{H \times M}$ and $\boldsymbol{w}_{vis} \in R^H$. The full input to $LSTM_{txt}$ is then given by $x_t^2 = [\hat{I}, \boldsymbol{h}_t^1, \boldsymbol{h}_{t-1}^2]$. The final output of the LSTM decoder is $\boldsymbol{h}_t^2$.

To handle unseen lexical items from novel objects, we initialize the softmax layer ($\boldsymbol{W}_p$ and $\boldsymbol{b}_p$) of the Up-Down model using parameters from the ELMo softmax layer ($\boldsymbol{W}_e$ and $\boldsymbol{b}_e$) and keep them fixed during training. Note that the dimensions of output layer ($\boldsymbol{W}_p$ and $\boldsymbol{b}_p$) and the LSTM output $\boldsymbol{h}_t^2$ are different. We add an additional fully connected layer

with a non-linearity function $tanh$ to project $\boldsymbol{h}_t^2$ to the dimension of $\boldsymbol{W}_p$. We have:

$$\boldsymbol{v}_t = tanh(\boldsymbol{W}_t \boldsymbol{h}_t^2 + \boldsymbol{b}_t) \tag{3.13}$$

$$P(y_t | y_{1:t-1}, I) = softmax(\boldsymbol{W}_p \boldsymbol{v}_t + \boldsymbol{b}_p) \tag{3.14}$$

where $W_t \in \mathbb{R}^{H^e \times H}$, $b_t \in \mathbb{R}^{H^e}$, H is the $LSTM_{txt}$ hidden dimension, $W_p \in \mathbb{R}^{H^e \times C}$, $b_p \in \mathbb{R}^C$ and C is the caption vocabulary size.

3.3.3 The Neural Baby Talk model

In this thesis, we use the NBT model with minor modifications for fair comparison with the Up-Down model. The Neural Baby Talk (NBT) model uses image objects $\mathbb{O}^{vg}$ and $\mathbb{O}^{oi}$. It has two modes of generating text: **Txt** Mode and **Vis** Mode. In **Txt** Mode, the NBT model is similar to the Up-Down model. In **Vis** Mode, the NBT model copies from image object labels in $\mathbb{O}^{oi}$. Note that these labels are often generic and can be represented by multiple words. For example, the label *boat* could be expressed as *ship* or *sailboat* in the captions. So we expand each image object label into a candidate word pool including its synonyms found in the training captions. Figure 3.5 shows an example of the two-step generation process of the NBT model:

- **Step 1:** The NBT model generates a hybrid template with concrete textual words (**Txt** Mode) and empty slots explicitly tied to specific image objects in $\mathbb{O}^{oi}$. In our example, the black words *a group of people riding on an* are all concrete textual words. The slot is tied to the object *elephant* in the blue box.
- **Step 2:** The NBT model fills these empty slots by selecting a particular word from the corresponding candidate word pool (which includes *elephant* and *calf* in our example) and the word's plurality (*singular* vs. *plural*) (**Vis** Mode). In our example, the NBT model selects *elephant* and *singular* as the refinement of the selected object label.

When training the NBT model, each ground truth caption is aligned with corresponding image objects. Words successfully aligned with object labels are seen as generated via **Vis Mode** and remaining words via **Txt Mode**.

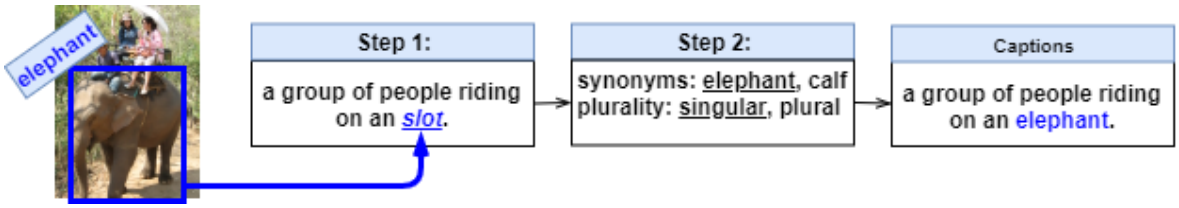

FIGURE 3.5: Two-step Generation Process of the NBT model.

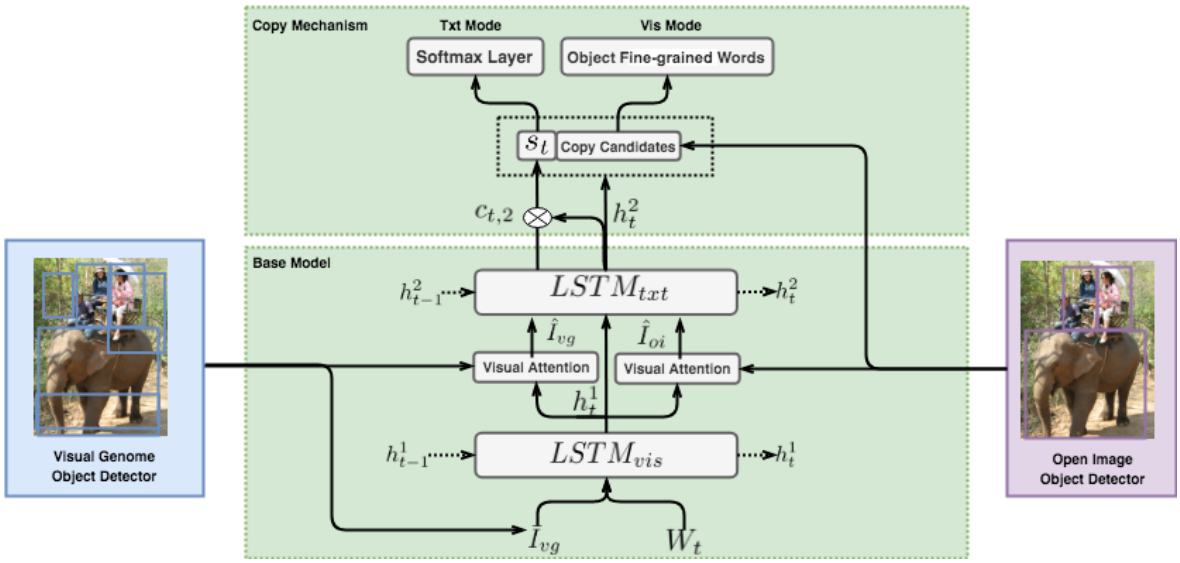

FIGURE 3.6: An overview of the NBT Model

NBT Base Model

Figure 3.6 shows an overview of the modified NBT model which shares a similar a two-layer LSTM architecture ($LSTM_{vis}$ and $LSTM_{txt}$, both of dimension M) with the above Up-Down model. The NBT model uses $\mathbb{O}^{vg}$ and $\mathbb{O}^{oi}$ image objects. In the NBT base model, all image objects are represented by *ROI vectors*, given by:

$$\mathbb{R}^* = \{\boldsymbol{r}_1^*, \boldsymbol{r}_2^*, \ldots, \boldsymbol{r}_{k_*}^*\}\,,\ \boldsymbol{r}_i^* \in \mathbb{R}^D \tag{3.15}$$

where $*$ can either be vg or oi.

In each step t, the input to $LSTM_{vis}$ is $\boldsymbol{x}_{t,1} = [\boldsymbol{w}_t, \bar{I}_{vg}]$ where $\boldsymbol{w}_t$ is the word embedding of previously generated word y_{t-1} and $\bar{I}_{vg}$ is the averaged *ROI vectors*, given by:

$$\bar{I}_{vg} = \frac{1}{k_{vg}} \sum_{i=1}^{k_{vg}} \boldsymbol{r}_i^{vg} \tag{3.16}$$

Given $\boldsymbol{h}_t^1$, the output of $LSTM_{vis}$, we have an attention module for $\mathbb{O}^{vg}$:

$$a_{t,i}^{vg} = \boldsymbol{w}_{vis}^T tanh(\boldsymbol{W}_r \boldsymbol{r}_i^{vg} + \boldsymbol{W}_h \boldsymbol{h}_t^1) \tag{3.17}$$

$$\bar{\boldsymbol{a}}_t^{vg} = softmax(\boldsymbol{a}_t^{vg}) \tag{3.18}$$

$$\hat{I}_{vg} = \sum_{i=1}^{k_{vg}} \bar{\boldsymbol{a}}_{t,i}^{vg} \cdot \boldsymbol{r}_i^{vg} \tag{3.19}$$

and a second attention module for $\mathbb{O}^{oi}$:

$$a_{t,i}^{oi} = \boldsymbol{w}_{vis}^T tanh(\boldsymbol{W}_r \boldsymbol{r}_i^{oi} + \boldsymbol{W}_h \boldsymbol{h}_t^1) \tag{3.20}$$

$$\bar{\boldsymbol{a}}_t^{oi} = softmax(\boldsymbol{a}_t^{oi}) \tag{3.21}$$

$$\hat{I}_{oi} = \sum_{i=1}^{k_{oi}} \bar{\boldsymbol{a}}_{t,i}^{oi} \cdot \boldsymbol{r}_i^{oi} \tag{3.22}$$

The input to $LSTM_{txt}$ is then $\boldsymbol{x}_{t,2} = [\boldsymbol{h}_t^1, \hat{I}_{oi} + \hat{I}_{vg}]$. The output and LSTM cell state of $LSTM_{txt}$ is h_t^2 and c_t^2 respectively. They are both used in the copy mechanism described below.

NBT Copy Mechanism

The NBT model has a two-stage copy mechanism given by:

$$p(y_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = p(y_t|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I}) p(r_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) \tag{3.23}$$

where $\boldsymbol{I}$ is the set of visual inputs to the NBT model. The first step estimates which object to copy via $p(r_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I})$. The second step selects a specific word given the selected objects r_t in the first step.

When copying from image object labels (**Vis** Mode), the NBT model uses objects from $\mathbb{O}^{oi}$ represented as *spatial features*, an *object label embedding* and *ROI vectors*. The spatial features include bounding box coordinates $(\bar{x}_{min}, \bar{y}_{min}, \bar{x}_{max}, \bar{y}_{max})$, width $\bar{w}_b$, height $\bar{h}_b$, area $\bar{w}_b \cdot \bar{h}_b$ and class confidence score $conf_b \in [0, 1]$. Box position and size features are all normalized by the input image width w_I and height h_I. Each spatial image feature vector thus has 8 elements, and we have:

$$\mathbb{S}^{oi} = \{\boldsymbol{s}_1^{oi}, \boldsymbol{s}_2^{oi}, \ldots, \boldsymbol{s}_{k_{oi}}^{oi}\}, \ \boldsymbol{s}_i^{oi} \in \mathbb{R}^8 \tag{3.24}$$

For the *object label embedding*, glove word embeddings $\boldsymbol{e}_i^{oi}$ of object labels are used. NBT first up-scales the *Spatial Features* $\boldsymbol{s}_i^*$ to dimension U (a model parameter) and then concatenates them with the other features, we have:

$$\boldsymbol{b}_i^{oi} = [\boldsymbol{W}_u \cdot \boldsymbol{s}_i^{oi} + \boldsymbol{b}_u,\ \boldsymbol{e}_i^{oi},\ \boldsymbol{r}_i^{oi}] \tag{3.25}$$

$$\mathbb{B}^{oi} = \{\boldsymbol{b}_1^{oi},\ \boldsymbol{b}_2^{oi},\ \ldots,\ \boldsymbol{b}_{k_{oi}}^{oi}\} \tag{3.26}$$

where $\boldsymbol{W}_u \in \mathbb{R}^{U\times 8}$ and $\boldsymbol{b}_u \in \mathbb{R}^U$. The attention scores over image objects in $\mathbb{O}^{oi}$ is $c_{t,i}^{oi}$ given by:

$$c_{t,i}^{oi} = \boldsymbol{w}_{txt}^T tanh(\boldsymbol{W}_b \boldsymbol{b}_i^{oi} + W_c h_t^2) \tag{3.27}$$

The choice of **Vis** Mode or **Txt** mode is determined by the attention score over image objects and a dummy object. The dummy object representation s_t is given by:

$$\boldsymbol{g}_t = \sigma(\boldsymbol{W}_x \cdot \boldsymbol{x}_{t,2} + \boldsymbol{W}_h \cdot \boldsymbol{h}_{t-1}^2) \tag{3.28}$$

$$s_t = \boldsymbol{g}_t \odot tanh(\boldsymbol{c}_t^2) \tag{3.29}$$

$$p(r_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = softmax([s_t, c_{t,1}^{oi} \ldots c_{t,k_{oi}}^{oi}]) \tag{3.30}$$

where $\boldsymbol{W}_x \in \mathbb{R}^{H\times D}$, $\boldsymbol{W}_h \in \mathbb{R}^{H\times H}$ and $\odot$ is element-wise product.

When **Txt** Mode is triggered, the NBT model generates a word from the caption vocabulary via a standard softmax layer, we have:

$$p(y_t|s_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = softmax(\boldsymbol{W}_{txt} \cdot \boldsymbol{h}_t^2 + \boldsymbol{b}_{txt}) \tag{3.31}$$

When **Vis** Mode is triggered, given the selected image object o_k^{oi} (represented by $\boldsymbol{b}_k^{oi}$), the NBT model does two independent predictions using current hidden states $\boldsymbol{h}_t^2$ and selected image object feature $\boldsymbol{b}_k^{oi}$: fine-grained word prediction to select a word from the corresponding candidate word pool (size l):

$$p_g(y_t^g|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = softmax(\boldsymbol{W}_g \cdot [\boldsymbol{h}_t^2, \boldsymbol{b}_k^{oi}] + \boldsymbol{b}_g) \tag{3.32}$$

where $\boldsymbol{W}_g \in \mathbb{R}^{K\times l}$, $\boldsymbol{b}_g \in \mathbb{R}^l$ and K is the size of $[\boldsymbol{h}_t^2, \boldsymbol{b}_k^{oi}]$. Plurality prediction (singular vs plural) is given by:

$$p_s(y_t^s|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = softmax(\boldsymbol{W}_s \cdot [\boldsymbol{h}_t^2, \boldsymbol{b}_k^{oi}] + \boldsymbol{b}_s) \tag{3.33}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{K\times 2}$, $\boldsymbol{b}_s \in \mathbb{R}^2$. Final word selection is then given by:

$$p(y_t|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = p_s(y_t^s|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I})p_g(y_t^g|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I}) \tag{3.34}$$

Note that original NBT model uses the COCO object label system and our newly proposed nocaps benchmark uses the Open Images object label system. Several fine-grained words in the original COCO candidate word pools are separate object classes (e.g. *man* and *woman* are fine-grained classes of *person* in the COCO benchmark). To adapt to this difference, we drop them as fine-grained words from the candidate word pools and retain them as Open Images object labels.

To train the NBT model with **Vis** Mode and **Txt** Model, it needs to align image object labels with caption words. The NBT model follows the rules below:

- The class prediction of the region proposal should be higher than 0.5.
- The Intersection over Union (IoU) of this region proposal with at least one of the ground truth bounding boxes is greater than 0.5.
- The predicted class is same as the object class of ground truth bounding box having highest IoU with this region proposal.

In this thesis, we drop the last rule because of the different image object labeling systems between COCO benchmark and Open Images benchmark, as explained above.

3.3.4 Constrained Beam Search

As introduced above, CBS is an inference-time beam search algorithm based on a Finite State Machine (FSM). In this thesis, a 24 state FSM is used to incorporate up to three selected objects as constraints, including two and three word phrases. Figure 3.7 shows the sub-FSM for two and three word multi-word expressions. The highest log-probability caption that satisfies at least two constraints is selected as the output caption.

The constraints come from image objects $\mathbb{O}^{oi}$. Note that not all image object labels are useful for captions, we propose simple but effective heuristic rules here to filter image objects. We first remove 39 object classes listed in Table A.4 from the constraint set, as these classes are either object parts or classes that we consider to be either too rare or too broad. We then resolve overlapping image objects (IoU $\geq$ 0.85) by removing the more

abstract of the two objects (e.g., *elephant* would suppress *mammal*) based on the Open Images class hierarchy (see Figure A.2 for details) (keeping both if equal). Finally, we take the top-3 image objects based on detection confidence as constraints.

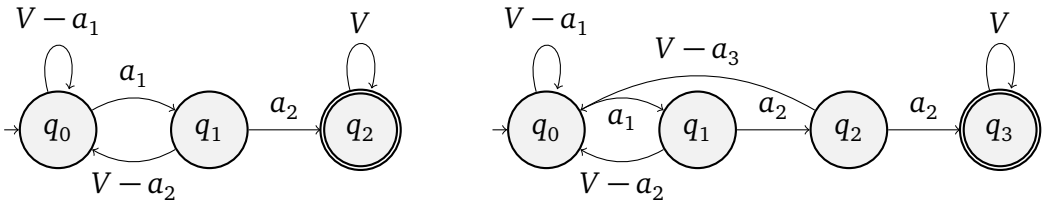

FIGURE 3.7: FSM for a two-word phrase (a_1 a_2) constraint (left) and a three-word phrase (a_1 a_2 a_3) constraint (right)

3.4 Experimental Results

In this section, the Up-Down and NBT models are evaluated on the nocaps benchmark. ELMo is applied to Up-Down model (+ ELMo) and constrained beam search is applied to both models (+ CBS). Ground truth image objects on nocaps images are also used for both models to establish a performance upper bound.

3.4.1 Model Setup

When applying ELMo in the Up-Down model, we use the ELMo full tensorflow checkpoint trained on the 1 Billion Word Language Model Benchmark[5] released in the official ELMo tensorflow repo[6]. Both Up-Down and NBT models are optimized by SGD [67]. We conduct hyper-parameter tuning on both models and choose the final model based on its performance on nocaps validation split. The hyper-parameters of LSTM decoders are shared by the Up-Down and NBT models. (See Table A.3).

[5] http://www.statmt.org/lm-benchmark/
[6] https://github.com/allenai/bilm-tf/

3.4.2 Results and Analysis

Ablation Study: ELMo and CBS

In this section, we quantify the effect of ELMo word representations and the impact of the simple constraint filtering heuristic. For ELMo, we compare it with two other word representations:

- Randomly initialized word representations (`Up-Down`), and
- The concatenation of Glove word embeddings [68] and dependency-based word embeddings [69], as proposed in [11] (`Up-Down` + GD).

For CBS constraints, we compare our proposed rules with three variants, including:

- Using all the object classes for constraints (w/o class),
- Using overlapping objects for constraints (w/o overlap), and
- Using no filtering heuristic at all (w/o both).

Note that in all cases we rank objects based on the confidence score for detected objects and pick the top-3 as the constraints.

As shown in Table 3.3, removing the 39 classes (See Table A.4) substantially improves the performance of constrained beam search and removing overlapping objects can also slightly improve the performance. This conclusion is consistent across the three models. It is clear that ELMo word representations work better than the other two representations, in particular for `out-of-domain` data.

The image object filtering rules above are heuristic. It is possible that we are overfitting some image object distributions in the `nocaps` validation set. In future work, we may improve this situation by developing a machine learning based image object filtering module. More analysis on the difference between prediction and ground truth image objects would need to be carried out.

COCO Performance Degradation As shown in row **2,3,4** and **6** of Table 3.4.2, there are substantial gains (almost 20 CIDEr for the `Up-Down` model and 2 CIDEr for the `NBT` model) in `nocaps` performance and corresponding large losses on COCO when we add constrained beam search. This may be because some of the image object labels from the

	In-Domain		Near-Domain		Out-of-Domain		Overall	
	C	S	C	S	C	S	C	S
Up-Down + w/o both	73.4	11.2	68.0	10.9	65.2	9.8	68.2	10.7
Up-Down + w/o class	72.8	11.2	68.6	10.9	65.5	9.7	68.6	10.8
Up-Down + w/o overlap	80.6	12.0	73.5	11.3	66.4	9.8	73.1	11.1
Up-Down +	80.0	12.0	73.6	11.3	66.4	9.7	73.1	11.1
Up-Down + GD + w/o both	72.8	11.2	68.4	10.8	66.3	9.8	68.6	10.7
Up-Down + GD + w/o class	72.3	11.2	68.6	10.9	66.9	9.7	68.8	10.7
Up-Down + GD + w/o overlap	77.0	12.0	73.5	11.4	67.2	9.7	72.8	11.1
Up-Down + GD	77.0	12.0	73.6	11.4	69.5	9.7	73.2	11.1
Up-Down + ELMo w/o both	73.3	11.5	68.6	10.9	70.0	10.8	69.6	10.8
Up-Down + ELMo w/o class	73.5	11.5	69.2	11.0	69.9	9.9	70.0	10.9
Up-Down + ELMo w/o overlap	79.8	12.3	73.7	11.4	72.0	9.9	74.2	11.2
Up-Down + ELMo	**79.3**	**12.4**	**73.8**	**11.4**	**71.7**	**9.9**	**74.3**	**11.2**
Human	**83.3**	**13.9**	**85.5**	**14.3**	**91.4**	**13.7**	**87.1**	**14.1**

TABLE 3.3: We investigate the effect of different object filtering strategies in Constrained Beam Search and report the model performance in nocaps eval data. We find that using both strategies with the ELMo model performs best. **C** stands for CIDEr and **S** stands for SPICE.

Open Images object detector, which are forced to be used by CBS, are not mentioned in the COCO ground truth captions. Limiting this degradation in the captioning setting is a potential focus for future work.

Language Models Help To handle novel vocabulary, CBS requires representations for novel words. We compare using ELMo encoding (row **3**) with the setting in which word embeddings are only learned during COCO training (row **2**). Note that in this setting the embedding for any word not found in COCO is randomly initialized. Surprisingly, the trained embeddings perform on par with the ELMo embeddings for the in-domain and near-domain subsets, although the model with ELMo performs much better on the

		COCO val 2017					nocaps val							
		Overall					In-Domain		Near-Domain		Out-of-Domain		Overall	
	Method	B-1	B-4	M	C	S	C	S	C	S	C	S	C	S
(1)	Up-Down	**77.0**	**37.2**	**27.8**	**116.2**	**21.0**	77.6	11.6	58.4	10.4	32.3	8.3	55.8	10.2
(2)	+ CBS	73.3	32.4	25.8	97.7	18.7	80.0	12.0	73.6	11.3	66.4	9.7	73.1	11.1
(3)	+ ELMo + CBS	72.4	31.5	25.7	95.4	18.2	79.3	12.4	73.8	11.4	71.7	9.9	74.3	11.2
(4)	+ ELMo + CBS + G	-	-	-	-	-	84.2	12.6	82.1	11.9	86.7	10.6	83.3	11.8
(5)	NBT	72.2	31.5	25.3	94.1	18.0	62.6	10.0	52.7	9.4	51.8	8.6	54.0	9.3
(6)	+ CBS	70.2	28.2	25.1	92.8	18.1	62.1	10.1	58.3	9.4	62.4	8.9	60.2	9.5
(7)	+ CBS + G	-	-	-	-	-	62.4	10.1	59.7	9.5	64.9	9.1	62.3	9.6
(8)	Human	66.3	21.7	25.2	85.4	19.8	**84.4**	**14.3**	**85.0**	**14.3**	**95.7**	**14.0**	**87.1**	**14.2**

TABLE 3.4: Single model image captioning performance on the COCO and nocaps validation sets. **B-1** and **B-4** stand for Bleu-1 and Bleu-4 respectively. **M** stands for Meteor. **C** stands for CIDEr and **S** stands for SPICE.

out-of-domain subset. This indicates: **i)** The large-scale COCO benchmark allows the model to effectively learn to use seen lexical items. **ii)** The pretrained language model, ElMo, has a positive impact on generating fluent captions with unseen words.

Better Object Detectors Help To evaluate the importance of object detectors, we supply ground truth object annotations to our full models (rows **4** and **7**). Note that ground truth object annotations undergo the same constraint filtering as predicted ones, except they are sorted by area rather than confidence. Comparing to prediction-reliant models (rows **3** and **6**), we see large gains on all splits for our Up-Down model (around 9 CIDEr and around 0.6 SPICE), but lesser gains for the NBT model. As image object detectors improve, we expect to see commensurate gains on nocaps benchmark performance.

Potential for Further Improvement As shown in Table 3.4.2, when decoding with beam search, the Up-Down model outperforms NBT by 15.0 CIDEr and 1.6 SPICE in in-domain and by 5.7 CIDEr and 1.0 SPICE in near-domain. This indicates that the Up-Down model learns to generate more fluent and meaningful captions than the NBT model when image objects are presented during training. In addition, the NBT model

	In-Domain		Near-Domain		Out-of-Domain		Overall					
Method	C	S	C	S	C	S	B-1	B-4	M	R	C	S
Up-Down	73.7	11.6	57.2	10.3	30.4	8.1	74.1	18.9	22.9	50.7	54.5	10.1
+ ELMo + CBS	76.0	11.8	74.2	11.5	66.7	9.7	76.6	18.4	24.4	51.8	73.1	11.2
NBT	62.8	10.3	51.9	9.4	48.9	8.4	71.8	14.2	21.8	48.0	54.3	9.4
+ CBS	61.9	10.4	57.3	9.6	61.8	8.6	69.6	12.4	21.6	46.7	59.9	9.5
Human	**80.6**	**15.0**	**84.6**	**14.7**	**91.6**	**14.2**	**76.6**	**19.5**	**28.2**	**52.8**	**85.3**	**14.6**

TABLE 3.5: Single model image captioning performance on the nocaps test sets. **R** stands for Rouge. **B-1** and **B-4** stand for Bleu-1 and Bleu-4 respectively. **M** stands for Meteor. **C** stands for CIDEr and **S** stands for SPICE.

demonstrates its clear advantage in out-of-domain, surpassing the Up-Down model by 19.5 CIDEr and 0.3 SPICE. The copy mechanism in the NBT model indeed learns a good policy to mention novel object labels when generating captions. A clear pathway to further improve performance in nocaps is to combine the advantages of the Up-Down model (fluent captions with trained words or phrases) and the NBT model (captions mentioning novel objects). It may be that the copy mechanism in the NBT model hurts the quality of generated in-domain captions. Our next step (in Chapter 4) investigates a modified version of the NBT model that can generate fluent in-domain captions as well as the Up-Down model does.

"If I have seen further it is by standing on the shoulders of Giants"

Isaac Newton

4

UpDown-C: A New Novel Object Captioner

In Chapter 3, two state-of-the-art captioning systems, the Up-Down and NBT models are evaluated on the nocaps benchmark. Up-Down model and NBT model show their different advantages in in-domain and out-of-domain respectively. When decoding with Constrained Beam Search (CBS), the Up-Down model outperforms NBT by a large margin. Yet, we believe that the NBT model has great potential as it is trained to handle novel object mentions in the captions. The CBS only enforces the model to output novel object labels during inference time without enriching model's internal representations of novel objects.

Motivated by the above, we propose a new captioning model UpDown-C. Our first step is to combine the strengths from previous state-of-the-art systems and further improve nocaps benchmark performance.

In this chapter[1], we will describe our new UpDown-C model in terms of *what to copy* and *when to copy*. Experimental results show that UpDown-C outperforms the Up-Down and NBT models by a large margin when using ordinary beam search. The UpDown-C model even beats the Up-Down with CBS baseline without using CBS. Finally, UpDown-C model sets the new state-of-the-art on the nocaps benchmark. .

4.1 Model Design: UpDown-C

Figure 4.1 shows the overview of UpDown-C model architecture. The UpDown-C model uses the Up-Down model as backbone. The copy mechanism of UpDown-C is inspired by the NBT model: Each training caption is aligned with the corresponding image objects and each image label has a corresponding candidate word pool. Comparing with the NBT model, we simplify the process of deciding fine-grain words and include all synonym and plural words of each image label in its single candidate word pool.

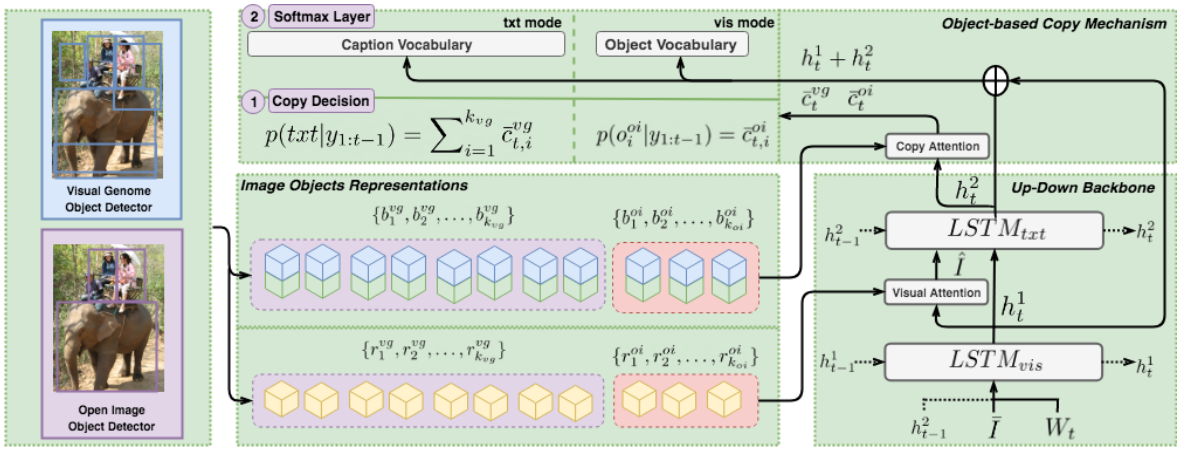

FIGURE 4.1: The UpDown-C model architecture.

Similar to the NBT model, the UpDown-C model performs the following steps (see Figure 4.2):

[1]Note that the material in this chapter has not been published yet. I developed the UpDown-C model under the supervision of Prof. Mark Johnson. Dr. Ian Wood also contributed to the UpDown-C model design.

- **Template Generation:** The NBT model generates a hybrid template with concrete textual words (**Txt** Mode) and empty slots explicitly tied to specific image objects in $\mathbb{O}^{oi}$. In our example, the black words *a group of people riding on an* are all concrete textual words. The slot is tied to the object *elephant* in the blue box.
- **Slot Filling:** Completing those empty slots by selecting a specific word from the corresponding candidate word pool. In this step, the UpDown-C model selects *elephant* from all possible variations (*elephant*, *elephants*, *calf* and *calves*).

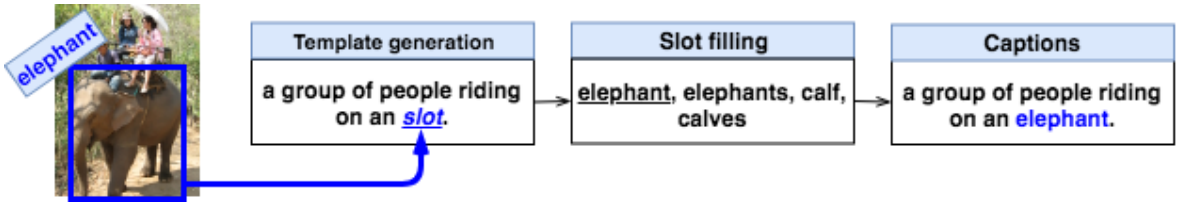

FIGURE 4.2: The two-step Generation Process for the UpDown-C model.

To demonstrate how our UpDown-C model operates over input images, figure 4.3 visualises the attention weights for each step in our UpDown-C model when generating the caption for an image in out-of-domain. Our UpDown-C model learns a reasonable alignment between caption words and image objects from the image-caption training data.

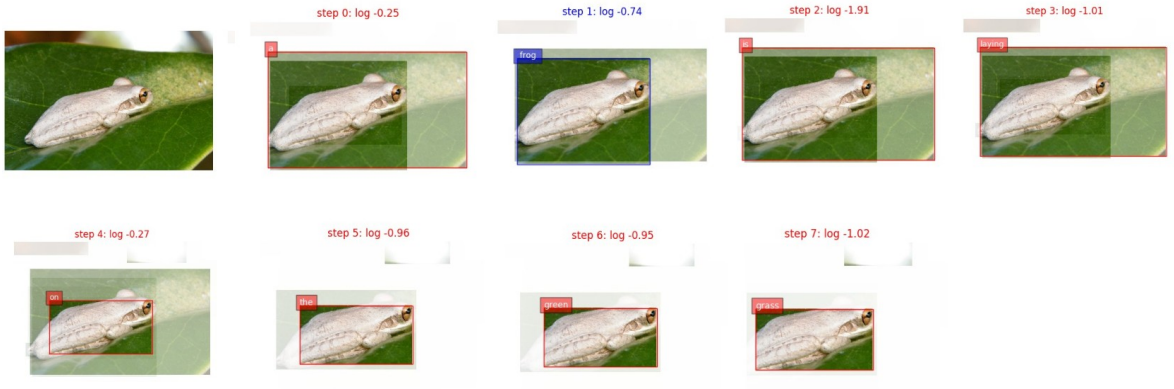

FIGURE 4.3: Attention Weights for each step when generating *A frog is laying on the green grass*. The red box indicates words generating from **Txt** Mode and the blue box indicates words generating from **Vis** Mode.

In the remainder of this section, we introduce details of the UpDown-C model, including the Up-Down *backbone* (Section 4.1.2), *what to copy* (Section 4.1.3), *when to copy* (Section 4.1.4) and training loss (Section 4.1.5).

4.1.1 UpDown-C Inputs

The UpDown-C model uses the same sets image objects $\mathbb{O}^{vg}$ and $\mathbb{O}^{oi}$ as the NBT model to represent input images. They are used in **i)** the Up-Down Backbone, represented by *ROI vectors* (all from the object detector O_{vg}) and **ii)** copy decision inputs, represented by *Spatial Features* (from both object detectors) and *Textual features*. Unlike the NBT model, UpDown-C uses all features from both $\mathbb{O}^{vg}$ and $\mathbb{O}^{oi}$ together.

4.1.2 Up-Down Backbone

The Up-Down model is used as backbone for the UpDown-C model. It is a two-layer LSTM model ($LSTM_{vis}$ and $LSTM_{txt}$, both of dimension M) with a visual attention module between the LSTM layers. Note that, unlike the original Up-Down model, here the visual attention utilises objects from both $\mathbb{O}^{vg}$ and $\mathbb{O}^{oi}$.

ROI vectors The *ROI vectors* here are the same as the those described in Chapter 3. We have:

$$\mathbb{R}^* = \{\boldsymbol{r}_1^*, \boldsymbol{r}_2^*, \ldots, \boldsymbol{r}_{k_*}^*\}\ ,\ \boldsymbol{r}_i^* \in \mathbb{R}^D\ ,\ k_* = k_{vg} + k_{oi} \tag{4.1}$$

where $*$ indicates object sources, either vg or oi.

Generation As the input to generation step t, different from the ELMo-enhanced Up-Down model which uses linear-weighted ELMo outputs, we only use the pre-trained ELMo [16] character encoder to represent the previously generated word y_{t-1} and averaged ROI vectors over image objects in $\mathbb{O}^{vg}$ and $\mathbb{O}^{oi}$:

$$W_t = ELMo_{ch}(y_{t-1}) \tag{4.2}$$

$$\bar{I} = \frac{\sum_{i=1}^{k_{vg}+k_{oi}} \boldsymbol{r}_i^*}{k_{vg} + k_{oi}} \tag{4.3}$$

The input to $LSTM_{vis}$ is $x_t^1 = [\bar{I}, h_{t-1}^2, W_t]$ where $h_{t-1}^2 \in \mathbb{R}^M$ is the previous hidden state from $LSTM_{txt}$. Visual input to $LSTM_{txt}$ is provided via an attention mechanism over ROI features. Given the output h_t^1 of $LSTM_{vis}$, we have:

$$a_{t,i}^* = \boldsymbol{w}_{vis}^T tanh(\boldsymbol{W}_r \boldsymbol{r}_i^* + \boldsymbol{W}_h h_t^1) \tag{4.4}$$

$$\bar{\boldsymbol{a}}_t^* = softmax(\boldsymbol{a}_t^*) \tag{4.5}$$

$$\hat{I} = \sum_{i=1}^{k_{vg}+k_{oi}} \bar{\boldsymbol{a}}_{t,i}^* \cdot \boldsymbol{r}_i^* \tag{4.6}$$

where $\boldsymbol{W}_r \in R^{H \times D}$, $\boldsymbol{W}_h \in R^{H \times M}$ and $\boldsymbol{w}_{vis} \in R^H$. The full input to $LSTM_{txt}$ is $x_t^2 = [\hat{I}, h_t^1, h_{t-1}^2]$. A residual connection is added to the LSTM decoder and the final output h_t is given by:

$$h_t = h_t^1 + h_t^2 \tag{4.7}$$

4.1.3 What objects to copy?

As shown in Figure 4.4, the image objects that are aligned with captions have an imbalanced distribution which leads to a bias towards copying frequent objects (i.e., *people*) and overlooks rare objects (i.e., *dolphin*). To counteract this bias, we discard the most frequent objects in $\mathbb{O}^{oi}$ before aligning object labels and caption words. That means, we force the `UpDown-C` model to learn to generate those excluded frequent image objects via the caption vocabulary softmax layer. Although this further decreases the fraction of words copied, our ablation study results show that removing frequent objects forces `UpDown-C` to learn to select rare objects and improves the overall performance.

4.1.4 When to copy?

This section introduces how `UpDown-C` makes decisions about when to copy from image object labels. We will first introduce the image object representations and then talk about our novel copy mechanism.

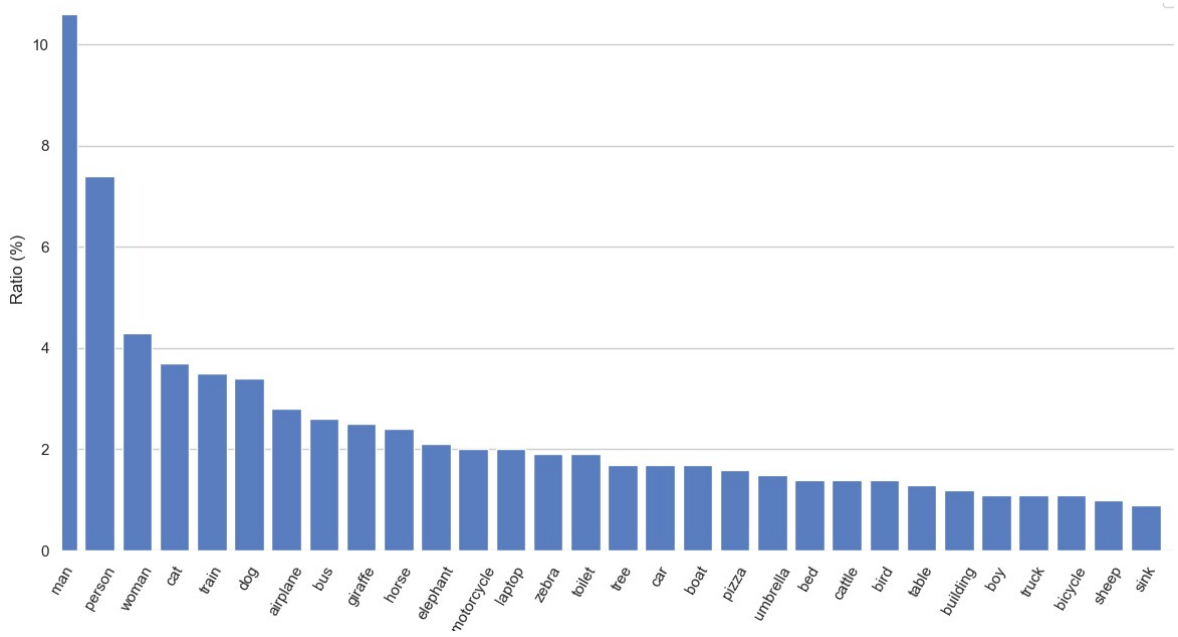

FIGURE 4.4: Open Images Object Distribution

Image objects Representations for Copy

Spatial Features The spatial features used here is the same with the ones in the NBT model (see Sectoin 3.3.3). We extract spatial features for both sets of objects. We have:

$$\mathbb{S}^* = \{\boldsymbol{s}_1^*, \boldsymbol{s}_2^*, \ldots, \boldsymbol{s}_{k_*}^*\}\,,\ \boldsymbol{s}_i^* \in \mathbb{R}^8 \tag{4.8}$$

Textual Features When copying words relating to image objects in $\mathbb{O}^{oi}$, textual information about these image objects is necessary for UpDown-C to make the good decisions. The labels of image objects in $\mathbb{O}^{vg}$ are not the copy candidates and we only assign a dummy value for them, which is later used as an indicator for **Txt** Mode.

The obvious representations of textual information for image objects include word embeddings of image object labels (i.e., Glove [68]) and *ROI features* of the image objects themselves (used in the NBT model). However, our experimental results show that using either word embeddings or ROI features results in poor generalization to unseen or rarely seen objects. We hypothesise that word embeddings or ROI features contain too much information specific to individual objects in the training data. The model memorizes these details and tends to select these objects over objects that are unseen during training.

Inspired from [70], we group object labels into more general concepts derived from the Open Images class hierarchy (see Figure A.2 for details). Figure 4.5 shows an example sub-tree in the Open Images class hierarchy with the selected nodes.

The Open Images class hierarchy groups together object classes that have similar properties and linguistic distribution. Representing these leaf nodes using their common ancestors minimizes the gap between seen and unseen image objects, enhancing the generalization ability of UpDown-C. To avoid training data bias, those selected sub-trees should have similar number of objects mentioned in the training captions. We start from the root node and gradually break each large tree node down to smaller ones. For example, in Figure 4.5, the “Vehicle” node is split into “Land vehicle” and “Vehicle” (child nodes from “Aerial vehicle” and “Watercraft” are represented by “Vehicle”). "Elephant" is represented by "Animal". We may also select leaf nodes if they have reasonable size. We refer this representations as *Abstract Embedding*. Each selected sub-tree is represented by a trainable dense embedding $e_i^* \in \mathbb{R}^A$ in UpDown-C model.

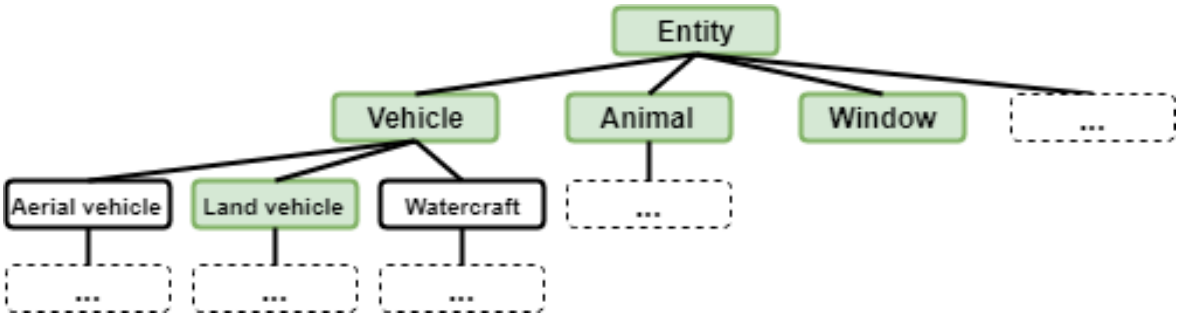

FIGURE 4.5: Open Images Class Hierarchy. Green Boxes are the selected entities.

To represent each image object using *Spatial features* and *Textual features*, we first map *Spatial Features* to a higher dimension and then concatenate the resulting feature vectors with *Textual Features* vector e_i^*. The final representation for each image is $\mathbb{B}^*$, given by

$$b_i^* = [W_u \cdot s_i^* + b_u,\ e_i^*] \tag{4.9}$$

$$\mathbb{B}^* = \{b_1^*,\ b_2^*,\ \ldots,\ b_{k_*}^*\} \tag{4.10}$$

where $W_u \in \mathbb{R}^{U \times 8}$ and $b_u \in \mathbb{R}^U$.

Object-Based Copy Mechanism

In this component, we propose to use image object information to decide whether to generate a word via standard way (**Txt Mode**) or copy from an object (**Vis Mode**).

Copy Decision Figure 4.6 compares the copy systems used in [13], [15] and UpDown-C model. We note that instead of using hidden states to make binary decisions (**Vis Mode** vs. **Txt Mode**), NBT uses a set of objects to decide which object to copy. UpDown-C model follows this trend and use the in-domain and out-of-domain image objects $\mathbb{O}^{oi}$ and $\mathbb{O}^{vg}$ to make copy decision. This encourages UpDown-C model to make decisions using diverse information and learn more complicated decision functions than linear ones to better handle rare or unseen image object cases.

$$\boldsymbol{c}_{t,i}^{vg} = \boldsymbol{w}_{txt}^{T} tanh(\boldsymbol{W}_b \boldsymbol{b}_{vg}^{i} + \boldsymbol{W}_c h_t^2) \tag{4.11}$$

$$\boldsymbol{c}_{t,i}^{oi} = \boldsymbol{w}_{txt}^{T} tanh(\boldsymbol{W}_b \boldsymbol{b}_{oi}^{i} + W_c h_t^2) \tag{4.12}$$

where $W_b \in R^{H \times B}$, $W_h \in R^{H \times M}$ and $w_{txt} \in R^H$. They are trainable during the training. We normalize two sets of attention scores $\boldsymbol{c}_t^{vg}$ and $\boldsymbol{c}_t^{oi}$ together using *softmax*, given by:

$$[\bar{\boldsymbol{c}}_t^{vg}; \bar{\boldsymbol{c}}_t^{oi}] = softmax([\boldsymbol{c}_t^{vg}; \boldsymbol{c}_t^{oi}]) \tag{4.13}$$

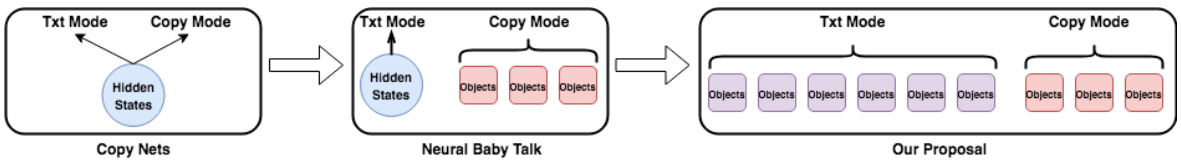

FIGURE 4.6: Copy Decision Comparison

UpDown-C model uses $\bar{\boldsymbol{c}}_t^{vg}$ and $\bar{\boldsymbol{c}}_t^{oi}$ to determine the mode in generation step t. UpDown-C copy the labels from out-of-domain image objects $\mathbb{O}_{oi}$ and we interpret the attention weights of each object in $\mathbb{O}^{oi}$ as the probability of copying that particular object. Whilst the attention weights of the in-domain objects $\mathbb{O}^{vg}$ are the support of in-domain "normal" mode and we calculate the probability of "normal" mode as the sum of the

attention weights over $\mathbb{O}^{vg}$. We have:

$$p(o_i^{oi}|\boldsymbol{y}_{1:t-1}) = \bar{\boldsymbol{c}}_{t,i}^{oi} \tag{4.14}$$

$$p(\text{txt}|\boldsymbol{y}_{1:t-1}) = \sum_{i=1}^{k_{vg}} \bar{\boldsymbol{c}}_{t,i}^{vg} \tag{4.15}$$

***Txt* Mode** UpDown-C uses the residual output h_t to generate words via the standard softmax layer (W_e, b_e). The vocabulary in this layer comes from COCO training captions and is initialized with the parameters in ELMo softmax layer and kept fixed during training. We use a fully-connected layer to adapt the different shape between ELMo softmax layer and LSTM hidden vectors, we have:

$$s_t = tanh(W_s \cdot h_t + b_s) \tag{4.16}$$

$$p(y_t|\text{txt}, \boldsymbol{y}_{1:t-1}) = softmax(W_e \cdot s_t + b_e) \tag{4.17}$$

***Vis* Mode** Given the selected object o_s^{oi} for copying, UpDown-C picks a specific word as the final output word. Similar to [15], each object label l corresponds to a few concrete words U_l. We construct U_l by finding object label l to plurals and synonyms in COCO training vocabulary. UpDown-C uses $h_t^1 + h_t^2$ to select a word from U_l via another softmax layer (W_l, b_l), given by:

$$f_t = ReLU(W_g \cdot h_t + b_g) \tag{4.18}$$

$$p(f_t|o_s^{oi}, \boldsymbol{y}_{1:t-1}) = softmax(W_l \cdot f_t + b_l) \tag{4.19}$$

where W_l and b_l is initialized with parameters from ELMo softmax layer.

Inference Constraints

In the nocaps benchmark, many captions mention ground truth image object labels. To encourage UpDown-C model to copy from object labels, we increase the probability of copying object labels by a constant factor β to $\bar{\boldsymbol{\alpha}}_{t,oi}$ just before the final decision only during the inference stage. We have:

$$p(\mathscr{O}_{oi}|\boldsymbol{y}_{1:t-1}) = \bar{\boldsymbol{\alpha}}_{t,oi} \cdot \beta \tag{4.20}$$

In addition, we constraint words shared by both caption vocabulary and object labels to only be generated once. We maintain the generation history during the inference and mask out objects or words when they have been generated.

4.1.5 Training Objective

Given the model parameter θ, we design two loss functions: $L_g(\boldsymbol{\theta})$ is to train the model to generate correct words when $g_t^* = \text{txt}$ and select correct objects when $g_t^* = \text{vis}$. $L_f(\boldsymbol{\theta})$ is to train UpDown-C to pick the correct specific word after the object is selected, given by:

$$L_g(\boldsymbol{\theta}) = -\sum_{t=1}^{T} \log\Big(p(y_t^*|\text{txt}, \boldsymbol{y}_{1:t-1}^*)p(\text{txt}|\boldsymbol{y}_{1:t-1}^*)\mathbb{1}_{(g_t^*=\text{txt})} + \big(\frac{1}{q_{oi}}\sum_{i=1}^{q_{oi}} p(o_i^{oi,c^*}|\boldsymbol{y}_{1:t-1}^*)\big)\mathbb{1}_{(g_t^*=\text{vis})}\Big) \tag{4.21}$$

$$L_f(\boldsymbol{\theta}) = -\sum_{g_t^*=\text{vis}} \log\Big(p(f_t^*|\mathbb{O}^{oi,c^*}, \boldsymbol{y}_{1:t-1}^*)\Big) \tag{4.22}$$

where y_t^* is the t^{th} word in the ground truth caption and g_t^* is the generation mode for it, $\mathbb{O}^{oi,c^*} = \{o_1^{oi,c^*}, o_2^{oi,c^*}, \ldots, o_{q_{oi}}^{oi,c^*}\}$ is the set of q_{oi} grounding objects that are aligned with y_t^* and f_t^* is the ground truth specific word to pick if $g_t^* = \text{vis}$, $\mathbb{1}_{(g_t^*=\text{txt})}$ is the indicator function which equals to 1 if y_t^* is textual word and 0 otherwise. We minimize the sum of the above two losses $L(\boldsymbol{\theta})$ in a multi-task learning fashion:

$$L(\boldsymbol{\theta}) = L_f(\boldsymbol{\theta}) + L_g(\boldsymbol{\theta}) \tag{4.23}$$

4.1.6 Constrained Beam Search

We are still interested to see whether constrained beam search can further improve the performance of the UpDown-C model. In our preliminary experiments we found that instead of controlling specific image objects to be used in the captions, a two-state finite state machine, as shown in Figure 4.7, was just as good as the previously used eight-state machine. Here q_0 is the starting state and q_1 is the accepting state where at least one of the image objects is used. The constraint $\mathbb{D} = \{D_1, D_2, D_3\}$ is similar to the lexical constraint

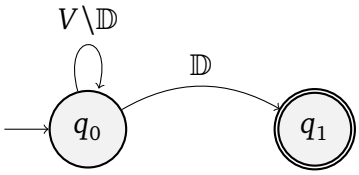

FIGURE 4.7: Two-state FSM for constraint $\mathbb{D} = \{D_1, D_2, D_3\}$

used in Chapter 3. In the UpDown-C model, image objects are viewed as a set of special output vocabularies.

4.2 Experiments

4.2.1 Experimental Setup

The training scheme and LSTM hyper-parameters for the UpDown-C model is the same with the Up-Down model described in Chapter 3. The dimension of the sub-tree trainable embedding $A = 1000$. The up-scale dimension for *Textual Features* $U = 1000$. The inference constant factor $\beta = 2.5$.

4.2.2 Main Results

Ablation study

In this ablation study, we show the impact of following components for UpDown-C. Table 4.1 shows the results of above experiments.

- **Using Objects for Copy Decision Helps** Unlike NBT model, we use two sets of image objects, $\mathbb{O}_{oi}$ and $\mathbb{O}_{vg}$ as the basic unit for copy mechanism. **- New Copy** uses the ordinary copy mechanism in NBT model. As shown in row **2**, UpDown-C improves the performance by 0.5 CIDEr and 1.8 CIDEr in near-domain and in-domain respectively and maintains similar performance in SPICE. UpDown-C can make more concrete decisions using image objects.
- **Filtering Image Objects Helps** Before training UpDown-C, we remove frequent image objects from aligning with caption words. **- Filtering** uses all detected image

		In-Domain		Near-Domain		Out-of-Domain		Overall	
	Method	C	S	C	S	C	S	C	S
(1)	Up-Down	80.7	11.9	74.4	11.3	77.7	10.6	75.9	11.3
(2)	- New Copy	80.8	12.0	73.9	11.3	75.9	10.5	75.3	11.3
(3)	- Filtering	78.7	11.7	70.4	11.0	73.5	10.1	72.2	10.9
(4)	- Embedding	80.7	12.0	73.3	11.3	73.2	10.4	74.3	11.2
(5)	- Object Bias	77.5	11.7	67.4	11.0	64.6	10.0	68.3	10.9
	Human	**84.4**	**14.3**	**85.0**	**14.3**	**95.7**	**14.0**	**87.1**	**14.2**

TABLE 4.1: The effect of imbalance removal components in UpDown-C

objects in $\mathbb{O}_{oi}$ to train UpDown-C. As shown in row **3**, UpDown-C improves the performance by 3.7 CIDEr and 0.4 CIDEr in overall and by 4.2 CIDEr and 0.5 SPICE in out-of-domain. That is, our UpDown-C model can learn to mention frequent image objects from **Txt** Mode well. This also allows UpDown-C to copy from rare image objects more often.

- **Abstract Embedding improves Generalization** In UpDown-C, we represent each object by one of its ancestors. **- Embedding** uses the concatenation of *ROI vectors* and word embeddings used in the NBT model. As shown in row **4**, UpDown-C improves the performance by 1.6 CIDEr and 0.1 SPICE in overall and by 4.5 CIDEr and 0.2 SPICE in out-of-domain. That shows that using the abstract embedding is helpful with novel objects. The knowledge learned from seen objects is successfully transformed to novel objects.
- **Bias Term as CBS** In the inference of UpDown-C, we add a bias term to the object probability and encourage UpDown-C to copy from image objects only during the inference. **- Object bias** removes bias term. As shown in row **5**, UpDown-C improves the performance by 7.6 CIDEr and 0.4 SPICE. This bias term can be viewed as a simple version of CBS with much less computational costs. It is interesting that although we add this term in every step, image object labels are added in the reasonable steps.

Comparing with existing systems

We compare UpDown-C performance on nocaps test set with results reported in [1] in Table 4.2. When decoding using ordinary beam search (see **BS**), UpDown-C clearly outperforms UpDown model by 19.9 CIDEr and 1.0 SPICE. Surprisingly, UpDown-C even outperforms Up-Down model with CBS by 1.4 CIDEr. That is, UpDown-C develops its own strategies of incorporating object information and we just need to inform UpDown-C to use object information, rather than explicitly instruct UpDown-C about which particular objects to use.

In addition, we conduct experiments that decode captions with CBS. We use the CBS with two-state FSM as discussed above (see Figure 4.7, **CBS**). Experiments show that the CBS decoding is still helpful to UpDown-C and improves UpDown-C by 0.4 CIDEr and 0.3 SPICE. Comparing with the existing systems, UpDown-C outperforms them by 2.0 CIDEr and 0.3 SPICE. That means the copy mechanism in UpDown-C can still be further improved to pick up those missed image objects.

Comparing with human performance, the UpDown-C model achieves better performance in Blue-1 and Blue-4 metrics and similar performance in ROUGE-L metric. All of these metrics measure the n-gram overlapping between generated sentences and ground truth sentences. Therefore, captions from UpDown-C model shares a similar lexical complexity with human captions. However, there are still large gaps in the other metrics, indicating UpDown-C captions can be improved in structural and semantic aspects.

Samples for the UpDown-C model

We show a few representative examples of the outputs from the UpDown-C model. Figure 4.8 shows three captions generated by the UpDown-C model, the Up-Down model with CBS and the NBT model with CBS. Among the three examples, the in-domain captions are similar to each other, whilst our UpDown-C model successful mentions the keywords *shotgun* and *cocktail* in the near-domain and out-of-domain captions.

		In-Dom.		Near-Dom.		Out-of-Dom.		Overall					
	Method	C	S	C	S	C	S	B-1	B-4	M	R	C	S
BS	Up-Down	73.7	11.6	57.2	10.3	30.4	8.1	74.1	18.9	22.9	50.7	54.5	10.1
	NBT	62.8	10.3	51.9	9.4	48.9	8.4	71.8	14.2	21.8	48.0	54.3	9.4
	UpDown-C	77.0	11.9	74.5	11.4	70.2	10.2	**77.2**	19.8	24.6	52.4	74.1	11.2
CBS	Up-Down	76.0	11.8	74.2	11.5	66.7	9.7	76.6	18.4	24.4	51.8	73.1	11.2
	NBT	61.9	10.4	57.3	9.6	61.8	8.6	69.6	12.4	21.6	46.7	59.9	9.5
	UpDown-C	78.0	12.2	75.2	11.6	69.2	10.3	77.0	**20.0**	24.8	52.5	74.5	11.5
	Human	**80.6**	**15.0**	**84.6**	**14.7**	**91.6**	**14.2**	76.6	19.5	**28.2**	**52.8**	**85.3**	**14.6**

TABLE 4.2: Comparison between UpDown-C and previous systems. **Bold** means the best performance given the same decoding conditions.

	in-domain	near-domain	out-of-domain
Method			
Up-Down + **CBS**	A beach with chairs and umbrellas and kites.	A man in a red hat holding a baseball rifle.	A wine glass on table with a bowl of food.
NBT + **CBS**	A beach with a bunch of umbrellas on a beach.	A baseball player holding a baseball rifle in the field.	A glass sitting on a table with food.
UpDown-C	A beach with umbrellas and kites on the beach	A man holding a shotgun and holding a baseball bat	A glass of orange cocktail on a white plate
Human	A couple of chairs that are sitting on a beach.	A man in a red hat is holding a shotgun in the air.	A cocktail in a glass with a piece of fruit.

FIGURE 4.8: Some challenging images from nocaps and the corresponding captions generated by the UpDown-C model, the Up-Down model and the NBT model.

"Everything is theoretically impossible, until it is done"

Robert Heinlein

5

Summary and Conclusions

The Image Captioning task, which connects vision and language, is an important building block for more sophisticated human-machine interaction systems. In this final chapter, we summarise the main contributions of this thesis and discuss some interesting directions for future work.

5.1 Summary

In this thesis, we first propose `nocaps`, a large-scale novel object captioning benchmark. This benchmark uses the existing COCO benchmark as training data and collects 11 human-generated captions for each of 15K selected images from the Open Images Dataset. Captions in `nocaps` contain more than 400 novel object classes (not seen in the COCO benchmark). The human annotators are encouraged to use ground truth image object

labels in the captions. We test two state-of-the-art captioning systems, the Up-Down NBT models on the nocaps benchmark. The empirical results show that **i)** the Up-Down model and the NBT model have different strengths; **ii)** Using a pre-trained language model is helpful for Novel Object Captioning; **iii)** Including image object labels using CBS can significantly improve the caption quality. This result motives us to develop the new captioning model UpDown-C incorporating strengths from the Up-Down and NBT models. We show that it is possible for a caption model to generate fluent in-domain captions and copy from unseen object labels simultaneously. The UpDown-C model outperforms the Up-Down model with CBS and sets the new state-of-the-art on the nocaps benchmark.

5.2 Future Directions

Finally, we conclude with a discussion of some promising research directions for the novel object captioning task.

Using Reinforcement Learning Reinforcement Learning has shown great success in image captioning tasks [71, 72]. However, these reinforcement learning methods use ground truth captions as the training rewards. In novel object captioning, we may not be given ground truth captions with novel objects. It is interesting to develop reference-less automatic metrics that can be incorporated into the existing reinforcement learning frameworks.

Using Non-Autoregressive Decoders Unlike Autoregressive decoders which generate text from left to right, non-autoregressive decoders generate text in arbitrary orders. The Middle-out decoder [73] can generate a fluent sentence by connecting one or two given keywords. It is interesting to explore the potential powers of Non-Autoregressive decoders for the Novel Object Captioning task. The novel object labels can be used as the starting keywords and the proposed decoder should learn to build up a fluent sentence based on those keywords.

A

Supplementary Materials

Info	ELMo	GPT	Bert	XLNet
Corpus	1 Billion Word Benchmark	BooksCorpus [74]	Wikipedia BooksCorpus	Wikipedia & BooksCorpus Giga5 [1] & ClueWeb 2012-B [2] Common Crawl [3]
Size	0.8B Tokens	1.1B Tokens	3.9B Tokens	32.9B Tokens

TABLE A.1: The Training Corpora for Large-scale Language Model.

[1] https://catalog.ldc.upenn.edu/LDC2011T07

[2] http://www.lemurproject.org/clueweb09/datasetInformation.php

[3] //http://commoncrawl.org

Benchmarks	Size	Collection	Data Source
COCO [4] [2]	995,684 Captions & 164,062 images	Human Annotation	Web
Flicker 8K [5] [75]	40,460 captions & 8,092 images	Human Annotation	Flicker
Flicker 30K [6] [76]	158,915 captions & 31,783 images	Human Annotation	Flicker
Im2Text [7] [77]	1M captions & 1M images	Automatic Collection	Flicker
Pinterest40M [8] [78]	300M captions & 40M images	Automatic Collection	Pinterest
Conceptual captions [9] [79]	3,369,218 captions & images	Automatic Collection	Web

TABLE A.2: Image Captioning Benchmarks.

Parameter	Value	Parameter	Value
Batch Size	150	Attention Size	768
LSTM Hidden Size	1200	Word Dropout	0.2
Image Feature	2048	ELMo Embedding	512
Learning Rate	0.015	Momentum	0.9
Clip Gradients	12.5	Weight Decay	0.001

TABLE A.3: Hyper-parameters for `Up-Down` and `NBT` models used in the experiment.

[4] http://cocodataset.org
[5] https://academictorrents.com/details/9dea07ba660a722ae1008c4c8afdd303b6f6e53b
[6] http://bryanplummer.com/Flickr30kEntities/
[7] http://www.cs.virginia.edu/~vicente/sbucaptions/
[8] https://github.com/mjhucla/P-Multimodal-Dataset-Toolbox
[9] https://ai.google.com/research/ConceptualCaptions

Parts		Too Rare or Too Broad	
Human Eye	Human Leg	Clothing	Building
Human Head	Human Beard	Footwear	Plant
Human Face	Human Body	Fashion Accessory	Land Vehicle
Human Mouth	Vehicle Registration Plate	Sports Equipment	Person
Human Ear	Wheel	Hiking Equipment	Man
Human Nose	Seat Belt	Mammal	Woman
Human Hair	Tire	Personal Care	Boy
Human Hand	Bicycle Wheel	Bathroom Accessory	Girl
Human Foot	Door Handle	Plumbing Fixture	
Human Arm	Skull	Tree	

TABLE A.4: Blacklisted object class names for constraint filtering (CBS) and visual word prediction (NBT).

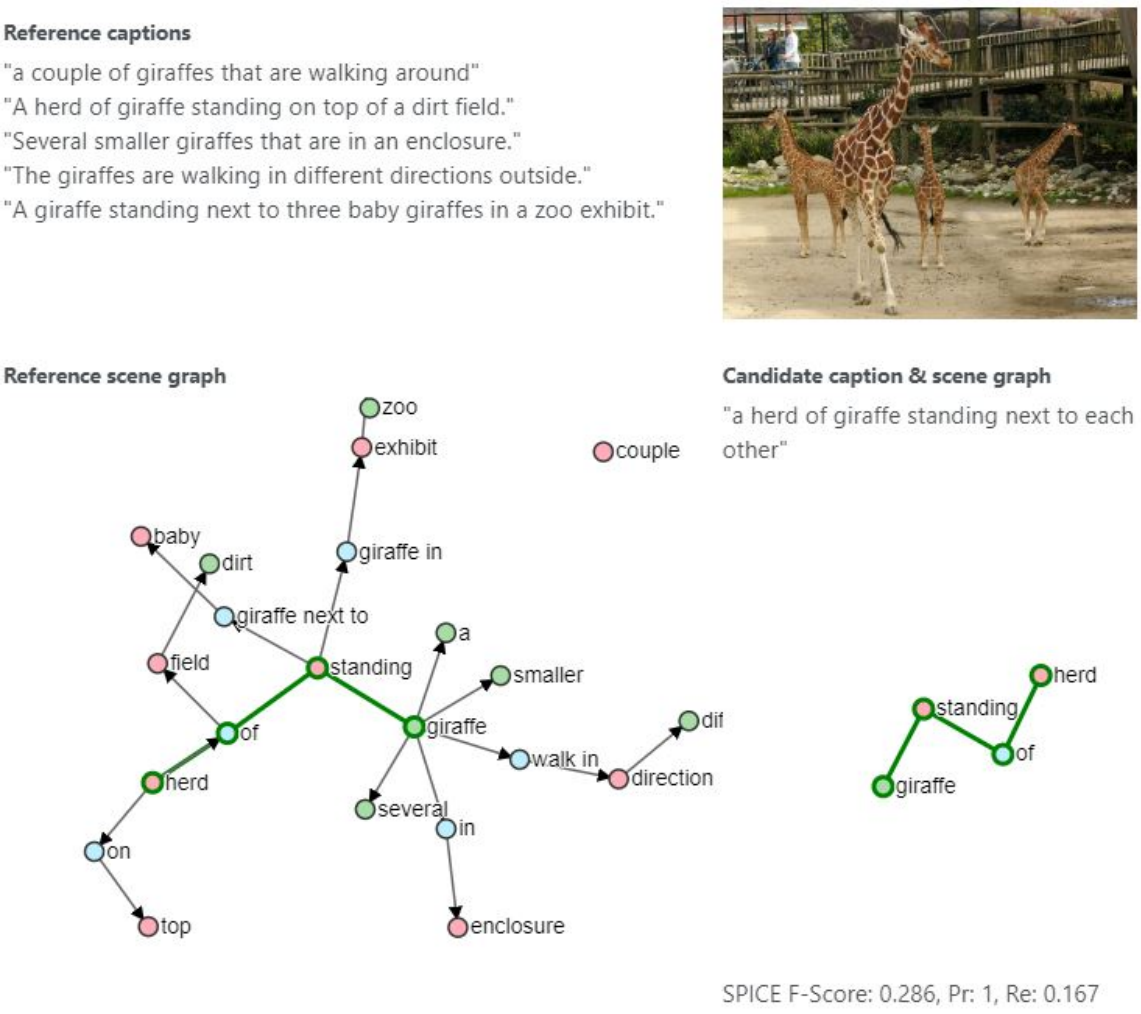

FIGURE A.1: An example of scene graph used in SPICE from https://panderson.me/spice/

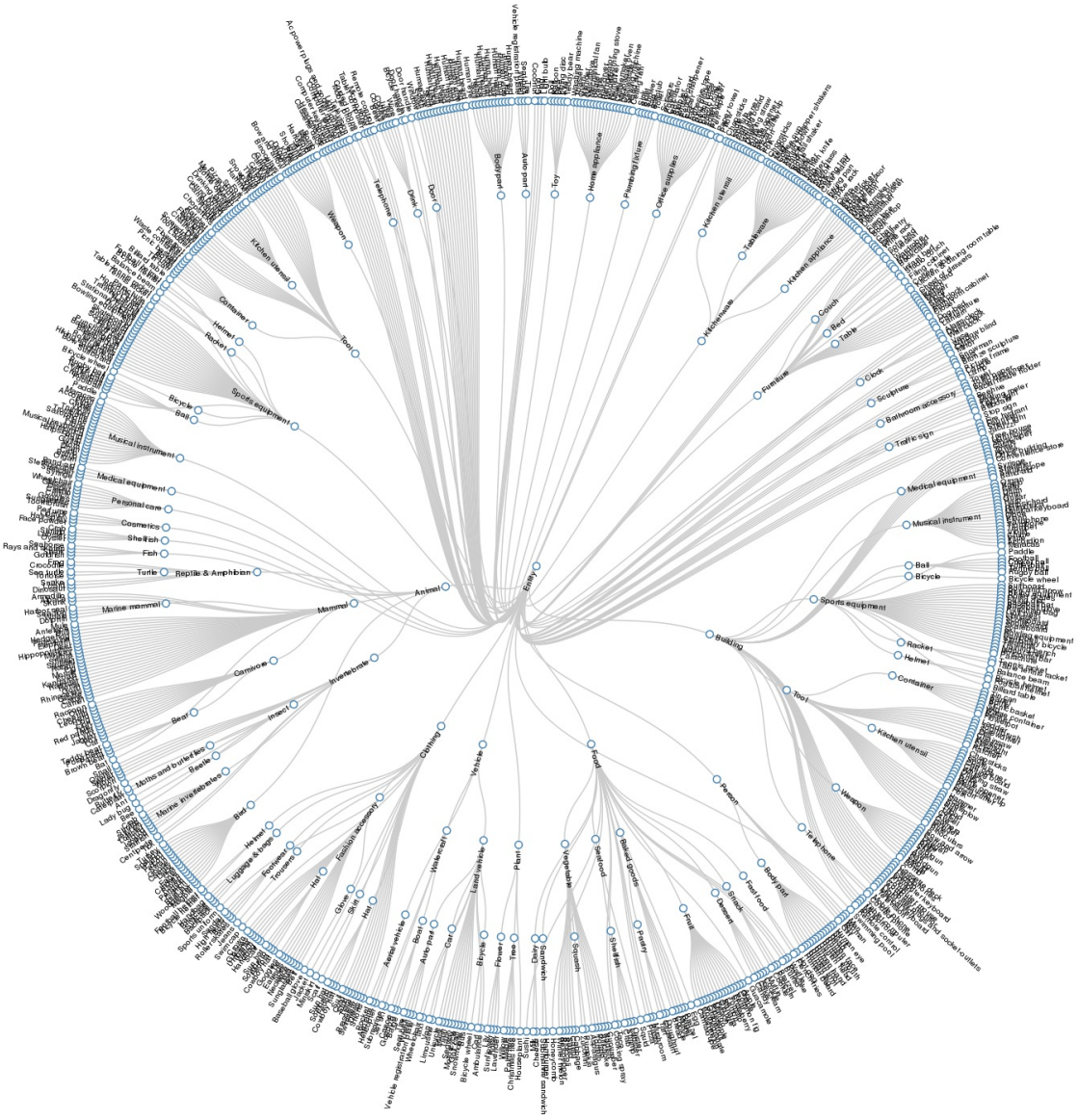

FIGURE A.2: An Overview of Open Images Class Hierarchy. Image is from https://storage.googleapis.com/openimages/2018_04/bbox_labels_600_hierarchy.json

References

[1] H. Agrawal, K. Desai, Y. Wang, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee, and P. Anderson. *nocaps: novel object captioning at scale*. ICCV (2019). v, 17, 47

[2] X. Chen, T.-Y. L. Hao Fang, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. *Microsoft COCO Captions: Data Collection and Evaluation Server*. arXiv preprint arXiv:1504.00325 (2015). vii, xiii, 17, 20, 52

[3] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, *et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale*. arXiv preprint arXiv:1811.00982 (2018). vii, 16, 17, 21

[4] M. A. Boden. *Mind as machine: A history of cognitive science* (Oxford University Press, 2008). 1

[5] K. Tran, X. He, L. Zhang, J. Sun, C. Carapcea, C. Thrasher, C. Buehler, and C. Sienkiewicz. *Rich image captioning in the wild*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2016). 2

[6] H. MacLeod, C. L. Bennett, M. R. Morris, and E. Cutrell. *Understanding blind people's experiences with computer-generated captions of social media images*. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp.

5988–5999 (ACM, New York, NY, USA, 2017). URL http://doi.acm.org/10.1145/3025453.3025814. 2

[7] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 2, 4, 13

[8] D. Parikh and K. Grauman. *Relative attributes*. In *2011 International Conference on Computer Vision*, pp. 503–510 (IEEE, 2011). 3

[9] T. Yao, Y. Pan, Y. Li, and T. Mei. *Incorporating copying mechanism in image captioning for learning novel objects*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 3, 13

[10] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei. *Pointing novel objects in image captioning*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 13

[11] P. Anderson, B. Fernando, M. Johnson, and S. Gould. *Guided open vocabulary image captioning with constrained beam search*. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 936–945 (Association for Computational Linguistics, Copenhagen, Denmark, 2017). URL https://www.aclweb.org/anthology/D17-1098. 3, 4, 14, 31

[12] O. Vinyals, M. Fortunato, and N. Jaitly. *Pointer networks*. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., *Advances in Neural Information Processing Systems 28*, pp. 2692–2700 (Curran Associates, Inc., 2015). URL http://papers.nips.cc/paper/5866-pointer-networks.pdf. 4, 10

[13] J. Gu, Z. Lu, H. Li, and V. O. Li. *Incorporating copying mechanism in sequence-to-sequence learning*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1631–1640 (Association for

Computational Linguistics, Berlin, Germany, 2016). URL https://www.aclweb.org/anthology/P16-1154. 4, 10, 42

[14] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to sequence learning with neural networks*. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., *Advances in Neural Information Processing Systems 27*, pp. 3104–3112 (Curran Associates, Inc., 2014). URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf. 4, 10

[15] J. Lu, J. Yang, D. Batra, and D. Parikh. *Neural baby talk*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 4, 13, 42, 43

[16] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. *Deep contextualized word representations*. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237 (Association for Computational Linguistics, New Orleans, Louisiana, 2018). URL https://www.aclweb.org/anthology/N18-1202. 5, 9, 38

[17] G. Tsoumakas and I. Katakis. *Multi-label classification: An overview*. International Journal of Data Warehousing and Mining (IJDWM) **3**(3), 1 (2007). 7

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *Imagenet classification with deep convolutional neural networks*. In *Advances in neural information processing systems*, pp. 1097–1105 (2012). 7

[19] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556 (2014). 7, 15

[20] 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*, pp. 770–778 (2016). 7, 15

[21] R. Girshick. *Fast r-cnn*. In *The IEEE International Conference on Computer Vision (ICCV)* (2015). 8

[22] S. Ren, K. He, R. Girshick, and J. Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks*. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., *Advances in Neural Information Processing Systems 28*, pp. 91–99 (Curran Associates, Inc., 2015). 8, 21

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. *Ssd: Single shot multibox detector*. In B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., *Computer Vision – ECCV 2016*, pp. 21–37 (Springer International Publishing, Cham, 2016). 8

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. *You only look once: Unified, real-time object detection*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 8

[25] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al. Learning representations by back-propagating errors*. Cognitive modeling **5**(3), 1 (1988). 9

[26] G. E. Hinton, J. L. McClelland, D. E. Rumelhart, *et al. Distributed representations* (Carnegie-Mellon University Pittsburgh, PA, 1984).

[27] J. L. Elman. *Finding structure in time*. Cognitive science **14**(2), 179 (1990). 9

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. *Distributed representations of words and phrases and their compositionality*. In *Advances in neural information processing systems*, pp. 3111–3119 (2013). 9

[29] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. *A neural probabilistic language model*. Journal of machine learning research **3**(Feb), 1137 (2003). 9

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of the 2019 Conference*

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019). URL https://www.aclweb.org/anthology/N19-1423. 9

[31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. *Language models are unsupervised multitask learners*. OpenAI Blog (2019). 9

[32] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. *Xlnet: Generalized autoregressive pretraining for language understanding*. arXiv preprint arXiv:1906.08237 (2019). 9

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. *Attention is all you need*. In *Advances in neural information processing systems*, pp. 5998–6008 (2017). 9

[34] E. Choi, H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang, and L. Zettlemoyer. *QuAC: Question answering in context*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2174–2184 (Association for Computational Linguistics, Brussels, Belgium, 2018). URL https://www.aclweb.org/anthology/D18-1241. 9

[35] Y. Wang, M. Johnson, S. Wan, Y. Sun, and W. Wang. *How to best use syntax in semantic role labelling*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5338–5343 (Association for Computational Linguistics, Florence, Italy, 2019). URL https://www.aclweb.org/anthology/P19-1529. 9

[36] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio. *Pointing the unknown words*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 140–149 (Association for Computational Linguistics, Berlin, Germany, 2016). URL https://www.aclweb.org/anthology/P16-1014. 10

[37] A. See, P. J. Liu, and C. D. Manning. *Get to the point: Summarization with pointer-generator networks*. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083 (Association for Computational Linguistics, Vancouver, Canada, 2017). URL https://www.aclweb.org/anthology/P17-1099. 10

[38] D. Elliott and F. Keller. *Image description using visual dependency representations*. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1292–1302 (2013). 11

[39] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III. *Midge: Generating image descriptions from computer vision detections*. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 747–756 (Association for Computational Linguistics, 2012). 11

[40] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. *Babytalk: Understanding and generating simple image descriptions*. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2891 (2013). 11

[41] A. Oliva and A. Torralba. *Building the gist of a scene: The role of global image features in recognition*. Progress in brain research **155**, 23 (2006). 11

[42] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick. *Exploring nearest neighbor approaches for image captioning*. arXiv preprint arXiv:1505.04467 (2015). 11

[43] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. *Every picture tells a story: Generating sentences from images*. In *European conference on computer vision*, pp. 15–29 (Springer, 2010). 11

[44] S. Hochreiter and J. Schmidhuber. *Long short-term memory*. Neural computation **9**(8), 1735 (1997). 11

[45] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. *Show, attend and tell: Neural image caption generation with visual attention*. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 2048–2057 (JMLR.org, 2015). URL http://dl.acm.org/citation.cfm?id=3045118.3045336. 11

[46] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. *Bottom-up and top-down attention for image captioning and visual question answering*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 12, 21, 23

[47] W. Wang, Z. Chen, and H. Hu. *Hierarchical attention network for image captioning*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8957–8964 (2019).

[48] N. Li and Z. Chen. *Image captioning with visual-semantic lstm*. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 793–799 (AAAI Press, 2018). URL http://dl.acm.org/citation.cfm?id=3304415.3304528. 12

[49] K. Papineni, S. Roukos, T. Ward, and W. Zhu. *Bleu: a method for automatic evaluation of machine translation*. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (2002). 12, 20

[50] A. Lavie and A. Agarwal. *Meteor: An automatic metric for MT evaluation with high levels of correlation with human judgments*. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL): Second Workshop on Statistical Machine Translation* (2007). 12, 20

[51] C. Lin. *Rouge: a package for automatic evaluation of summaries*. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) Workshop: Text Summarization Branches Out* (2004). 12, 20

[52] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. *Cider: Consensus-based image description evaluation*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). 12, 20

[53] P. Anderson, B. Fernando, M. Johnson, and S. Gould. *Spice: Semantic propositional image caption evaluation*. In B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., *Computer Vision – ECCV 2016*, pp. 382–398 (Springer International Publishing, Cham, 2016). 12, 20

[54] S. Venugopalan, L. Anne Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. *Captioning images with diverse objects*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 13

[55] Y. Wu, L. Zhu, L. Jiang, and Y. Yang. *Decoupled novel object captioner*. In *Proceedings of the 2018 ACM on Multimedia Conference (ACM MM)* (2018). 13

[56] C. Hokamp and Q. Liu. *Lexically constrained decoding for sequence generation using grid beam search*. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1535–1546 (Association for Computational Linguistics, Vancouver, Canada, 2017). URL https://www.aclweb.org/anthology/P17-1141. 14

[57] M. Post and D. Vilar. *Fast lexically constrained decoding with dynamic beam allocation for neural machine translation*. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1314–1324 (Association for Computational Linguistics, New Orleans, Louisiana, 2018). URL https://www.aclweb.org/anthology/N18-1119. 14

[58] P. Anderson, S. Gould, and M. Johnson. *Partially-supervised image captioning*. In *Advances in Neural Information Processing Systems*, pp. 1875–1886 (2018). 14

[59] A. P. Dempster, N. M. Laird, and D. B. Rubin. *Maximum likelihood from incomplete data*

via the EM algorithm. Journal of the royal statistical society. Series B (methodological) (1977). 14

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. In *CVPR09* (2009). 15

[61] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. *Visual genome: Connecting language and vision using crowdsourced dense image annotations*. International Journal of Computer Vision **123**(1), 32 (2017). URL https://doi.org/10.1007/s11263-016-0981-7. 16, 21

[62] D. Yadav, R. Jain, H. Agrawal, P. Chattopadhyay, T. Singh, A. Jain, S. B. Singh, S. Lee, and D. Batra. *Evalai: Towards better evaluation systems for ai agents*. arXiv preprint arXiv:1902.03570 (2019). 20

[63] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. *Speed/accuracy trade-offs for modern convolutional object detectors*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 21

[64] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. *Character-aware neural language models*. In *Thirtieth AAAI Conference on Artificial Intelligence* (2016). 22

[65] L. He, K. Lee, O. Levy, and L. Zettlemoyer. *Jointly predicting predicates and arguments in neural semantic role labeling*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 364–369 (Association for Computational Linguistics, 2018). URL http://aclweb.org/anthology/P18-2058. 23

[66] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. *GLUE: A multi-task benchmark and analysis platform for natural language understanding*. In *International Conference on Learning Representations* (2019). URL https://openreview.net/forum?id=rJ4km2R5t7. 23

[67] L. Bottou. *Large-scale machine learning with stochastic gradient descent*. In *Proceedings of COMPSTAT'2010*, pp. 177–186 (Springer, 2010). 30

[68] J. Pennington, R. Socher, and C. Manning. *Glove: Global vectors for word representation*. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543 (2014). 31, 40

[69] O. Levy and Y. Goldberg. *Dependency-based word embeddings*. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 302–308 (2014). 31

[70] M. Ciaramita and M. Johnson. *Supersense tagging of unknown nouns in wordnet*. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 168–175 (Association for Computational Linguistics, 2003). 41

[71] 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*, pp. 290–298 (2017). 50

[72] J. Gao, S. Wang, S. Wang, S. Ma, and W. Gao. *Self-critical n-step training for image captioning*. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 50

[73] S. Mehri and L. Sigal. *Middle-out decoding*. In *Advances in Neural Information Processing Systems*, pp. 5518–5529 (2018). 50

[74] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books*. In *arXiv preprint arXiv:1506.06724* (2015). 51

[75] 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**, 853 (2013). 52

[76] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. *Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models*. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649 (2015). 52

[77] V. Ordonez, G. Kulkarni, and T. L. Berg. *Im2text: Describing images using 1 million captioned photographs*. In *Advances in neural information processing systems*, pp. 1143–1151 (2011). 52

[78] J. Mao, J. Xu, Y. Jing, and A. Yuille. *Training and evaluating multimodal word embeddings with large-scale web annotated images*. In *NIPS* (2016). 52

[79] P. Sharma, N. Ding, S. Goodman, and R. Soricut. *Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565 (2018). 52