UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

LEARNING TO MERGE LANGUAGE AND VISION

A DEEP EVALUATION OF THE ENCODER, THE ROLE OF THE TWO MODALITIES, THE ROLE OF THE TRAINING TASK.

Ravi Shekhar

Advisor

Prof. Raffaella Bernardi

Università degli Studi di Trento

Co-Advisor

Prof. Raquel Fernández

University of Amsterdam

May 2019

Abstract

Most human language understanding is grounded in perception. There is thus growing interest in combining information from language and vision. Multiple models based on Neural Networks have been proposed to merge language and vision information. All the models share a common backbone consisting of an encoder which learns to merge the two types of representation to perform a specific task. While some models have seemed extremely successful on those tasks, it remains unclear how the reported results should be interpreted and what those models are actually learning. Our contribution is three-fold. We have proposed (a) a new model of Visually Grounded Dialogue; (b) a diagnostic dataset to evaluate the encoder ability to merge visual and language input; (c) a method to evaluate the quality of the multimodal representation computed by the encoder as general purposed representations. We have proposed and analyzed a cognitive plausible architecture in which dialogue system modules are connected through a common grounded dialogue state encoder. *Our in-depth analysis of the dialogues shows the importance of going beyond task-success in the evaluation of Visual Dialogues: the dialogues themselves should play a crucial role in such evaluation. We have proposed a diagnostic dataset,* FOIL *which consists of images associated with incorrect captions that the model has to detect and correct. Finally, we have used FOIL to evaluate the quality of the multimodal representation produced by an encoder trained on different multimodal tasks. We have shown how the training task used effects the stability of the representation, their transferability and the model confidence.*

Keywords

Language and Visual Models, Visually Grounded Dialogues, Diagnostic Datasets, Transfer Learning

Acknowledgments

I must thank many people sfor contributing to my wonderful journey during a Ph.D. student. First of all, I am deeply indebted to my advisor Prof. Raffaella Bernardi for her dedication, encouragement, motivation. I am in debt for her inspiring guidance and supervision throughout my thesis work. The best part of our numerous discussion was that she always looks for the positivity in the stupidest idea and results.

During my Ph.D. have been very fortunate to collaborate and learn from many remarkable people. I would like to especially thank Raquel Fernández for her unique perspective and insightful way of analysis. Towards the end of my thesis, I was fortunate to work with Barbara Plank. I am grateful to all my collaborators: Aurelie, Elia, Enver, Moin, Aashish, Tim, Sandro, Ece, Yauhen, for all discussions and comments.

I am grateful to the very efficient administrative department of the University of Trento for making life more comfortable, especially Andrea from ICT School to answer all the queries related to the Ph.D. program. I thank Welcome office for all the support and trips. I would also like to give a special thanks to the server administrators, specifically to Daniele from CiMeC, to have all servers working without that it would have been close to impossible to keep the deadlines.

I want to thank all my friends not only for their support during research also for making life at Trento memorable one. Sandro, Claudio, Alberto, Tin, Dul, Lam, Emanuele, Subhankar, Amol, thanks for making life enjoyable during my stay here.

I would also like to express my gratitude towards everyone in my family. It would not have been possible to even dream of studying further without their support. I would also like to especially thank my wife, Shiva, for all her support. This would not have been possible without her constant support.

Contents

List of Figures

List of Tables

Publications

This thesis collects several articles which have been published during my PhD. As such, most of the contents of this thesis have appeared in the following publications:

- Ravi Shekhar, Aashish Venkatesh, Tim Baumgärtner, Elia Bruni, Barbara Plank, Raffaella Bernardi, and Raquel Fernández, "**Beyond task success: A closer look at jointly learning to see, ask, and GuessWhat** ", *In Proc. of* 17^{th} *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) 2019(Long-Oral).* (Chapter 3)

- Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurelie Herbelot, Moin Nabi, Enver Sangineto and Raffaella Bernardi, "**FOIL it! Find One mismatch between Image and Language caption**", *In Proc. of* 55^{th} *Annual Meeting of the Association for Computational Linguistics (ACL), 2017 (Long-Oral).* (Chapter 4)

- Ravi Shekhar, Ece Takmaz, Raffaella Bernardi, and Raquel Fernández, "**Evaluating the Representational Hub of Language and Vision Models** ", *In Proc. of* 13^{th} *International Conference on Computational Semantics (IWCS), 2019 (Long-Oral).* (Chapter 5)

Other Publications

- Ravi Shekhar, Tim Baumgärtner, Aashish Venkatesh, Elia Bruni, Raffaella Bernardi, and Raquel Fernández, "**Ask No More: Deciding when to guess in referential visual dialogue**", *In Proc. of* 27^{th} *International Conference on Computational Linguistics (COLING), 2018.*

- Ravi Shekhar, Sandro Pezzelle, Aurelie Herbelot, Moin Nabi, Enver Sangineto and Raffaella Bernardi, "**Vision and Language Integration: Moving be-**

yond Objects", *In Proc. of* 12^{th} *International Conference on Computational Semantics (IWCS), 2017.*

- Sandro Pezzelle, Ravi Shekhar and Raffaella Bernardi, "**Building a bagpipe with a bag and a pipe: Exploring conceptual combination in Vision**", *In Proc. of* 5^{th} *Workshop on Vision and Language at ACL, Berlin, Germany, 2016.*

Chapter 1

Introduction

Human brains are multi-modal. Our knowledge comes from various types of inputs that we simultaneously receive and jointly process in parallel. In this direction, one of the goals of artificial intelligence (AI) researchers is to enable a computer to see, understand the visual concepts, express this understanding into a sentence, and reason about the understanding. To achieve this goal, the integration of language and vision (LaVi) information is essential such that the model can use knowledge from both modalities.

Ideally, a model that genuinely merges information received from different modalities should add and merge them to enrich the knowledge, and not merely correlate them based on the data. For instance, given an image representing a woman standing in front of a door and opening it (as shown in Figure 1.1), and the description *Mary is coming to the office*; by using one of the two inputs alone we can answer the questions *Is the door close?* (visual) and *Where is Mary coming?* (linguistic), but we can also answer the question *Was Mary's office close?* by merging the two modalities. Ideally, we would like the LaVi model to be able to reach this ability. To achieve this goal, the model has to be able to merge a fine-grained representation and knowledge from both modalities.

Toward this goal, multiple tasks have been proposed in the literature. For example, image captioning (Kulkarni et al., 2013; Hodosh et al., 2013a), visual

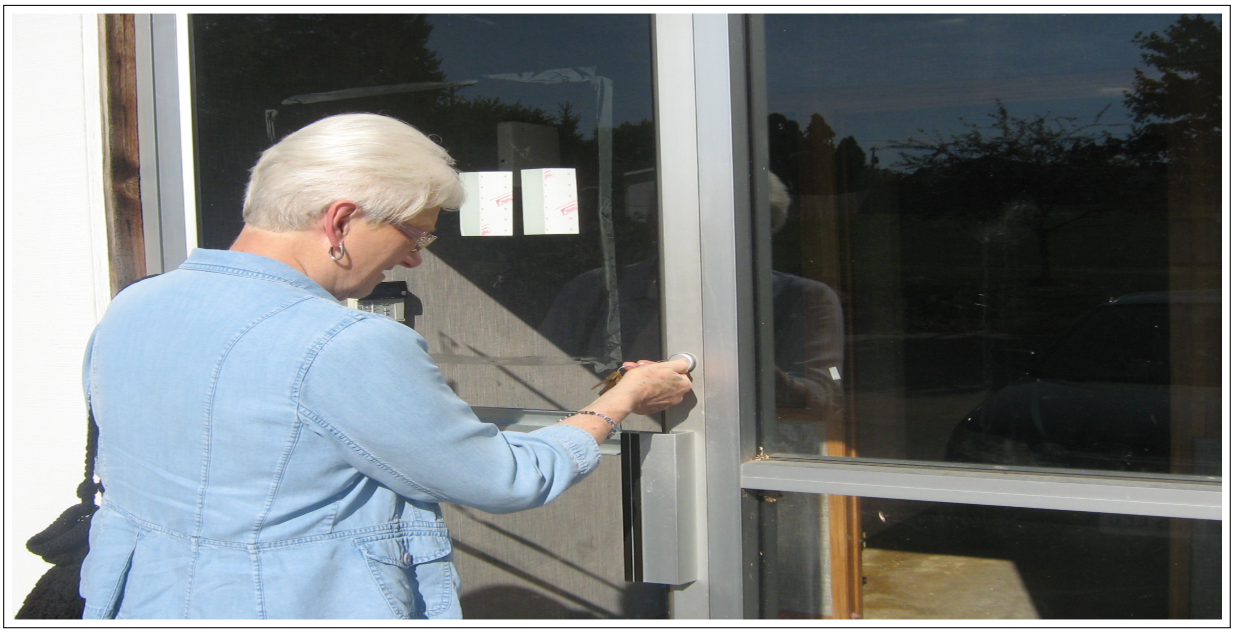


Figure 1.1: Caption: *Mary is coming to the office*. An example of complementary information from language and vision.

question answering (Antol et al., 2015), visual question generation (Mostafazadeh et al., 2016), visual reference resolution (Kazemzadeh et al., 2014), and visual dialogue (Das et al., 2017a), etc. These tasks are designed with the increasing level of complexity to merge the two modalities. For example, image captioning is mostly about model seeing the image and describing it in natural language. While the visual question answering (VQA) is asking a question about content of the image and visual dialogue is asking multiple coherent questions in a sequence about the image. To perform these tasks computational multi-modal systems using deep neural networks (DNN) have been proposed. In general, the visual information is processed using the convolutional neural network (CNN) (LeCun et al., 1998), pre-trained on the ImageNet (Krizhevsky et al., 2012). The textual information is processed using the recurrent neural network (RNN) (Mikolov et al., 2010; Hochreiter and Schmidhuber, 1997). RNNs are trained end-to-end on the task-specific dictionary. Using these visual and textual features, different fusion mechanisms been proposed: like concatenation (Vinyals et al., 2015), dot product (Antol et al., 2015), bilinear pooling (Park et al., 2016) etc. Broadly there are two types of fusion mechanisms: early and late fusion (Hodosh and Hockenmaier, 2016). In early fusion, features are merged at the word level. While in the late fusion, features are merged at the sentence level. Later, different attention mechanisms (Xu et al., 2015; Yang et al., 2015) are incorporated in the pipeline to achieve better-fused representations. Most of the attention mechanisms are focused on visual attention. Recently, memory network (Xiong et al., 2016; Chunseong Park et al., 2017) is also being used to improve performance further.

These models perform exceptionally well on the given task. However, it is not clear that these models merge the two modalities to complement each modalities information. The results presented in Zhou et al. (2015) and Hodosh and Hockenmaier (2016) show that the tasks currently proposed by the LaVi community can be tackled successfully by learning correlations within and between the two modalities (Zhou et al., 2015) and by extracting the gist of the image rather than understanding it in details (Hodosh and Hockenmaier, 2016). To explore this, there is a growing interest to understand these models, i.e. how to explain specific decisions made by models? In this direction, broadly three mechanisms are proposed: diagnostic dataset (Hodosh and Hockenmaier, 2016; Johnson et al., 2017), merged representation based model analysis (Kádár et al., 2015, 2017), gradient-based model analysis (Zhou et al., 2016; Selvaraju et al., 2017). Using the diagnostic datasets (Hodosh and Hockenmaier, 2016; Johnson et al., 2017), limitations of state-of-the-art (SoA) models are shown by doing minimal change in one of the input modalities. In merged representation based model analysis (Kádár et al., 2017; Conneau et al., 2018), models have been probed on the different linguistic and visual properties to analysis the representation learned. In the case of gradient-based model analysis (Selvaraju et al., 2017; Goyal et al., 2016b), the weights of the neurons are directly projected into the input space to analyze the effect of different characteristics of the models directly.

Against this background, we aim to provide a mean to evaluate whether the encoders of SoA LaVi models truly merge vision and language representations and to understand whether the task on which a LaVi model is trained effects the encoder performance in merging the two modalities. In particular, we compare tasks in which the model has to retrieve a linguistic answer (Visual Question Answering) vs. an image (using Referring Expression and Visual Dialogue).

1.1 Contributions of this study

Joint Model for Visual Dialogue We introduce a single visually grounded dialogue state encoder to jointly train the guesser and question generator modules to address a foundational issue on how to integrate visual grounding with dialogue system components. A decision-making module is also introduced in this joint setting to stop the dialogue when enough information is gathered to perform the task. We also do a first in-depth study to compare different learning approaches. Our study shows that the linguistic skills of the models differ dramatically, despite approaching comparable task success levels. This underlines the importance of linguistic analysis to complement solely task success based evaluation.

FOIL: a diagnostic dataset We have proposed a diagnostic dataset, called the FOIL dataset, and three tasks based on it. The FOIL dataset contains image captions pair such that each image is associated with a foiled caption and a good one. The foil caption is created by inserting a wrong (foil) word in the caption in place of the correct (target) word. The foil word can be of various parts of speech. We exploit available resources to obtain the target foil pairs (target::foil, i.e., a target word can be replaced by a foil word). Based on these target foil pairs, we replace one word at a time in the caption to create a foil caption.

We evaluated SoA LaVi models with the FOIL dataset against three tasks: (a) Classify the caption as good or wrong; (b) Localize the mistake; (c) Correct the mistake. These tasks are designed in such a way that, they will evaluate the strengths and weaknesses of models in their ability to merge language and vision representation.

Encoder Evaluation Multiple LaVi tasks have been proposed. However, it is not clear how well these tasks are fusing/encoding the information from language and vision. Is the merged representation similar for all tasks? Also, are these merged representations good enough to be transferred to another task?

In this direction, we investigate the three popular LaVi tasks: VQA, ReferIt and GuessWhat and scrutinize the multimodal representations a model learns through them. First of all, we proposed a common dataset for these tasks by minimizing the difference as much as possible. At the task level, we re-design the tasks to have common evaluation protocol. We evaluate the merged representations on the FOIL dataset with classification tasks. Using the FOIL classification task, we evaluate how well the learned representations can be transferred from three tasks to the FOIL task. We also evaluate the learned representation‘s semantic space using the Representation Similarity Analysis and Nearest Neighbour overlap of the object representation. We find that the merged representation of ReferIt and GuessWhat is similar compared to the VQA. We also find that all encoders give more weight to the visual input than the linguistic one.

1.2 Thesis Outline

The remaining sections of this thesis are organized as follows:

In Chapter 2, we provide an overview of LaVi tasks, dataset and models proposed. The first contribution of this doctoral study, a novel method for merging language and vision for the visual dialogue is described in Chapter 3. In Chapter 4, we describe the second contribution as the FOIL dataset and tasks to show the limitations of the SoA LaVi models. The last contribution on encoder evaluation described in Chapter 5. Finally, in Chapter 6 we conclude this dissertation by some remarks.

Chapter 2

Related Work

Recently, there has been a growing interest in combining information from language and vision (LaVi). The interest is based on the fact that many concepts can be similar in one modality but very different in the other, and thus capitalizing on both information turns out to be very effective in many tasks, for example, image captioning, visual question answering, visual dialogue, etc. All these tasks provide unique opportunities to understand the commonality and uniqueness of modalities. In this chapter, we will review different LaVi tasks, dataset, and corresponding models.

2.1 Non-Interactive Tasks

Multiple non-interactive tasks have been proposed for LaVi. In this section, we will review three of those tasks: Image Captioning, Referring Expression, and visual question answering.

2.1.1 Image Captioning

In the image captioning (IC) task, given an image, the system has to describe the content of the image, see Figure 2.1 for an example. In general IC task is formulated in two ways; as a retrieval task and generation task. In IC retrieval

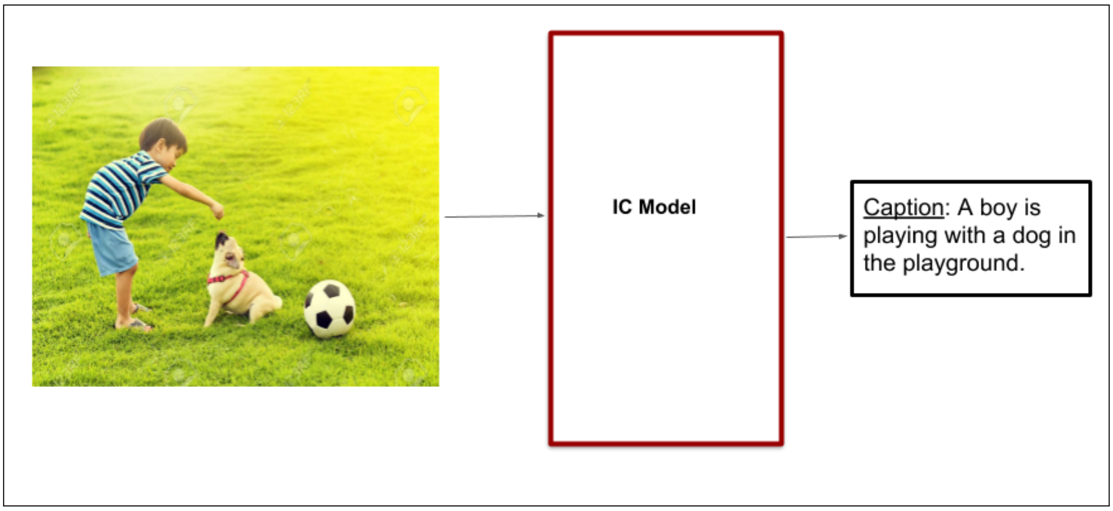


Figure 2.1: An example of image captioning system. For a given image IC model has to generate a corresponding caption.

task (Hodosh et al., 2013a), given an image model has to retrieve the closest description of the image from the set of possible description. In IC generation task (Kulkarni et al., 2013; Fang et al., 2015; Chen and Lawrence Zitnick, 2015; Donahue et al., 2015; Karpathy and Fei-Fei, 2015; Vinyals et al., 2015; Wang et al., 2016), the goal is to generate a caption for a given image, such that it is both semantically and syntactically correct, and properly describes the content of that image.

Dataset A large number of image captioning datasets has been proposed for both natural scene (Rashtchian et al., 2010; Elliott and Keller, 2013; Hodosh and Hockenmaier, 2013; Plummer et al., 2015; Lin et al., 2014a; Krishna et al., 2017) and abstract scene (Zitnick and Parikh, 2013; Zitnick et al., 2013). In this thesis, we focused on the natural scene dataset. The Pascal1K (Rashtchian et al., 2010) is one of early large scale image captioning dataset, based on 1000 images from the Pascal 2008 object recognition dataset (Everingham et al., 2010). For each image, five descriptions are collected by humans using Amazon Mechanical Turk (AMT). Similar to the Pascal1K, the Flickr8K (Hodosh and Hockenmaier, 2013) and Flickr30K dataset (Plummer et al., 2015) collected five descriptions per image using AMT, using approximately 8,000 and 30,000 images from the Flickr respectively.

The MS-COCO dataset (Lin et al., 2014a), one of the widely used image captioning dataset, consist of 123,287 images. All the images in the MS-COCO contain at least one of 80 object categories. On average, there are more than 3.5 objects present in an image. Each image is annotated with bounding box over the objects, and objects are segmented. Each image is annotated with a minimum of five captions per image using ATM. Multiple datasets are created on top of the MS-COCO dataset. Our proposed FOIL dataset also uses this dataset, (see Section 4.2 for more details.)

A further level of annotation is proposed in the Visual Genome dataset (Krishna et al., 2017). Instead of only focusing on the objects in the image, the dataset provides detailed annotation of object interactions and attributes, by providing textual descriptions of multiple regions present in the images. The Visual Genome dataset is built using the images from the YFCC100M (Thomee et al., 2015) and MS-COCO (Lin et al., 2014a).

Model Image captioning models can be categorized into two broad categories based on space used for the feature: the single space model and the joint space model. In single space models, visual features are projected into the textual feature space or vice-versa and then different learning algorithms are applied to generate a good caption. In joint space models, both visual and textual features are projected into a common space, different from their own space, and then in this space captions are generated based on different learning algorithms.

Instead of describing the whole image in a sentence, earlier attempts have been made to associate the image regions to `nouns`. Different statistical models are proposed based by Barnard et al. (2003) on the joint distribution of image regions and words into the `single space`. Soft-hierarchical clustering is performed on the textual data to learn the mapping between the regions and the nouns. A multi-modal extension to a mixture of latent Dirichlet allocation models, which tries to learn explicit correspondence between regions and words was also used. Similarly, Duygulu et al. (2002) have formulated the object recognition problem as a machine translation task. Where they have to translate from the visual feature to the text feature. Different regions of the image are annotated with the word and mapping from a region feature to a word feature is modeled as an energy minimization problem.

Gupta and Davis (2008) proposed to use `preposition` and `comparative adjectives` also, not only nouns, for image classification task. Image regions respective positions, like above, below, are used to model prepositions and comparative adjectives, etc. The learning algorithm is formulated as an energy minimization problem into the `joint space`. To further improve the performance, post-processing is performed on the prediction based on frequency correctness and semantic identifications.

Farhadi et al. (2010) formulated the image captioning problem as generating triplet of $< object, action, scene >$. Based on these triplets, a `coherent sentence` is generated. In this work, joint modeling of the visual and textual feature is used to generate the common feature space for both. This common representation is used to retrieved sentences based on an image query and vice versa. Similar to this, Ordonez et al. (2011) also map both representations in common space. However, in this work, more natural images and sentences are used, and instead of focusing only on object, action, scene tripled as in Farhadi et al. (2010), they have also used different aspects of images like attributes of the image (e.g., furry cats).

Socher and Fei-Fei (2010) used a few labeled images along with large collection of the newspaper articles to learn the joint mapping of the image and sentences into the `joint space`. Instead of mapping whole images and sentences into the joint space, they mapped different image segments and words into the joint space using kernelized canonical correlation analysis on feature representations of both the visual and the textual domain. This is one of the early work to use `unlabeled data` to solve the problem in semi-supervised fashion.

Hodosh et al. (2013a) formulated image description as a `ranking task`. Where a given pool of captions is ranked based on the images. Instead of using one image one caption, they proposed to use `multiple captions` per image.

Li et al. (2011) also proposed to use unlabeled textual data from the web and use them to describe the image. In this work, firstly, the objects (like cat, dog) present in the image are detected along with their visual attributes (like black, furry) and then the spatial relationship (like under, above) between these objects are estimated. After generating, these different pieces of information, pre-trained n-grams on the web data is used to add the possible objects and attributes in the list. On this list, different fusion techniques are proposed to optimize the n-grams frequency to compose them into a sentence. Kulkarni et al. (2013) extended this approach and used the conditional random field(CRF) to predict the best caption for the image.

Most of the above the image captioning models are evaluated based on ranking tasks. One of the major drawbacks of the ranking based task is that it has limited vocabulary words. Based on the recent success of the recurrent neural network (RNN) on different language tasks, like machine translation Wu et al. (2016), different `generative models` are used for the image caption. In machine translation, an RNN model consists of an ‘encoder’ and a ‘decoder’ module. An ‘encoder’ module reads the source sentence and transforms it into a fixed-length vector representation, which in turn is used as the initial hidden state of a ‘decoder’ RNN module that generates the target sentence. Figure 2.2 describes the overall structure of the IC model based on the encoder-decoder structure. Most the recent models follow similar architecture, having one or

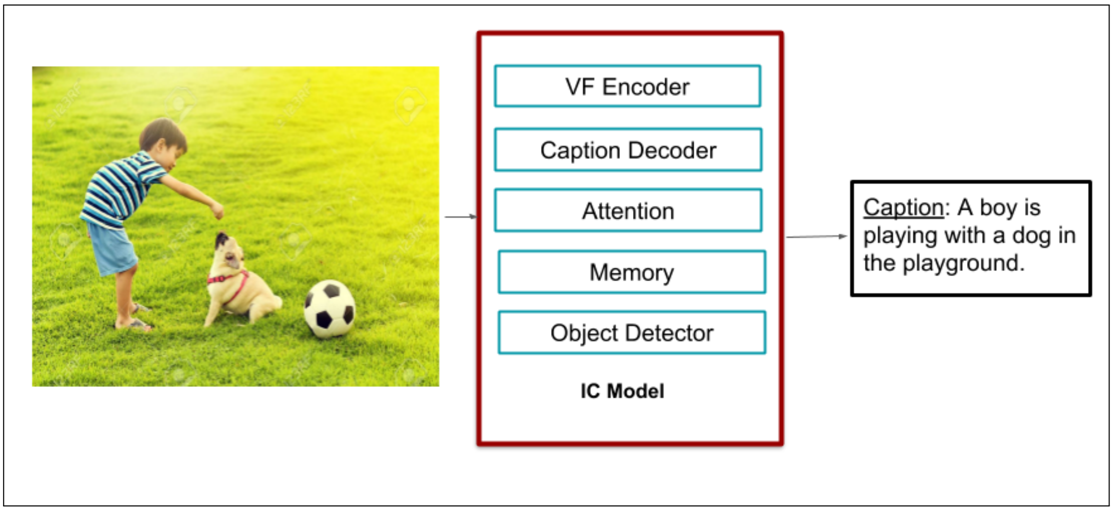


Figure 2.2: Overview Neural Network based IC Model as encoder-decoder.

more of these components like attending, object detector etc.

Instead of mapping the whole image and the corresponding description sentence into a joint embedding space, Karpathy et al. (2014) proposed to map the fragments of the images and the `fragments of the sentence` into a joint space. While optimizing the mapping, they have used structured max-margin optimization technique. The structure is provided in terms of the fragment of the sentence and the image. We have used this model for the generation of the FOIL dataset (see Section 4.2 for more details.) Similar to this, Vinyals et al. (2015) proposes an end to end image captioning model. In which, they take the image feature extracted based on convolutional neural network (CNN) (Krizhevsky et al., 2012) as the output of the encoder module of the RNN and feed this as an input to the decoder of the RNN. They optimize the model by maximizing the probability of the correct description given the image. For RNN, Long-Short Term Memory (LSTM) net is used.

Vendrov et al. (2015) learn the visual-semantic hierarchy of image and description of images over words, sentences, and images. Instead of preserving the distance between the visual semantic hierarchy and the embedding space, they preserve the partial order between the visual semantic hierarchy and the embedding space. With this approach, they not only generate good caption but are also able to perform different tasks like the composition of images objects.

Along with generative process, attempts are being made to use `attention` based mechanism (Xu et al., 2015; You et al., 2016) to generate better descriptions. Xu et al. (2015) proposes two different attention based models, based on hard and soft assignments, attending different salient parts of the image while generating the image description. To provide the attention, instead of using the last layer of CNN, the fully connected layer is used. You et al. (2016) uses attention mechanism to semantically fusing results obtained from the image to the word and the word to the image properties.

Anderson et al. (2018) used fine-grained object-level features based on Faster R-CNN (Girshick, 2015) on multiple image regions. They enrich the visual region feature with the object properties, like color, size, etc, to get better feature representation. Using enriched feature combined with attention is used to perform both IC and VQA task.

2.1.2 Refering Expression

To have a finer level of image understanding referring expression generation is proposed. In referring expression task for a given image and an object region in the image model has to generate a natural language referring expression for the object region.

Dataset Kazemzadeh et al. (2014) created one of the first large scale LaVi referring expression dataset, called Referit. The Referit dataset is created using two player ReferIt Game. Player 1 sees the image and a region and has to write the referring expression for that region. Player 2 sees the image and referring expression and has to select one of the objects in the image. Based on the majority agreement referring expressions and the corresponding region is selected. For the Referit dataset, images are taken from the ImageCLEF

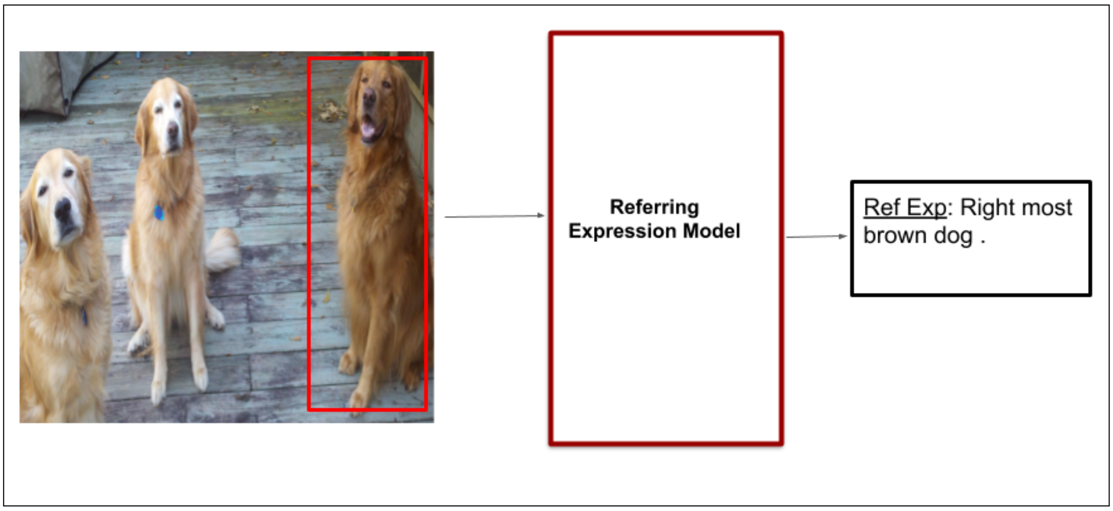


Figure 2.3: An example of Referring Expression. For a given image and bounding box of the target object (in red box), the model has to generate unambiguous referring expression for the target object.

dataset. Yu et al. (2016a) and Mao et al. (2016) simultaneously created the referring expression dataset using the images from MS-COCO dataset using the Referit game, called RefCOCO and RefCOCOg respectively (see Section 5.4). In general, the RefCOCO dataset consists of a concise description and shorter length, while the RefCOCOg dataset is comparatively slightly longer length description. We have used the RefCOCO dataset for the encoder evaluation, more details in Section 5.2

Model Most of the proposed model for the referring expression is inspired by IC models. Kazemzadeh et al. (2014) proposed a model as a mapping from the target object and image to referring expression based on the properties of the objects. Similar to Vinyals et al. (2015), Yu et al. (2016a) used image feature to initialize the RNN state to generate the referring expression. Instead of only using full image feature, they have extracted features of each object of the same type and fed the difference in the object feature and full image feature to LSTM to generate the referring expression. Mao et al. (2016) also follow a similar method, instead of using single RNN, they have used different RNN for the different regions. To generate discriminative sentences, discriminative maximum mutual information based loss is used. Further, performance is improved using data augmentation in a semi-supervised way. Nagaraja et al. (2016) use the context object regions of the region to improve the performance of the generated expression. Context regions are discovered using multiple instance learning. Liu et al. (2017) first learn the attribute of the objects and used these learned attributes to comprehend expression.

2.1.3 Visual Question Answering

Image captioning and referring expression generation suffers from evaluation protocol i.e., sensitive to n-grams (Anderson et al., 2016). Image captioning is also shown to provide a coarse level of image understanding. To overcome some of these, visual question answer (VQA) task (Malinowski and Fritz, 2014; Ren et al., 2015b; Gao et al., 2015; Antol et al., 2015; Yu et al., 2015; Zhu et al., 2016) is proposed. In VQA, for a given image one can ask any free-form and open-ended question about the image in natural language and system have to provide natural language answer to the question. The answer can be in the form of multiple choice, i.e., given $2-4$ choices the system has to provide which option is most likely to be the answer of the question or it can be in terms of fill in the blanks, where system need to generate appropriate word for the blank position.

Dataset The DAQUAR (Malinowski and Fritz, 2014), DAtaset for QUestion Answering, is one of the first datasets for VQA using natural images. The DAQUAR is build using images from the NYU-Depth v2 dataset (Silberman et al., 2012). Question/Answers pair is collected by both automatically and using human annotation. Ren et al. (2015b) automatically created the Question/Answers(QA) pairs based on the caption of MS-COCO images. Gao et al.

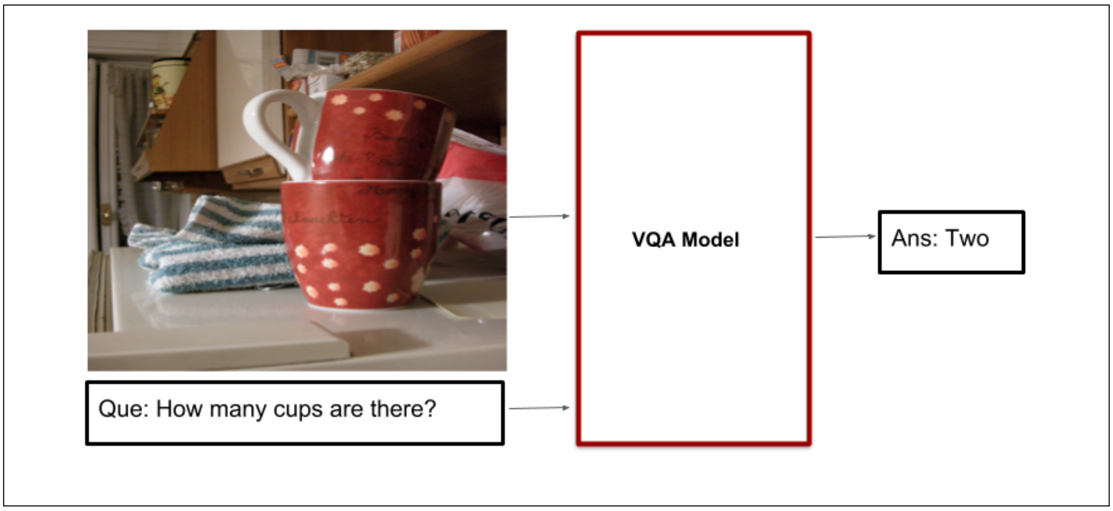


Figure 2.4: An example VQA. For a given image and corresponding question, model has to produce an answer.

(2015) also used the MS-COCO images, the QA pairs were collected using AMT. The QA pairs were first collected in Chinese, then converted into English by human translators.

Antol et al. (2015) created one of the widely used VQA dataset(VQAv1), using images from the MS-COCO dataset and human annotators collected QA pair. For each image on average at least 3 QA pairs were collected. The dataset also allows evaluation using a multiple-choice setting, by providing 18 candidate answers, of which only one is correct, for each question-image pair. The further balanced dataset in proposed by Goyal et al. (2016a).

The Visual Genome dataset (Krishna et al., 2017) also contains QA pair such that all the questions start with 'seven Ws' i.e., who, what, where, when, why, how, and which. Questions are collected such that without images, the answer cannot be guessed. The Visual7w (Zhu et al., 2016) dataset is a subset of the Visual Genome with additional annotations and evaluation using a multiple-choice setting, with four candidate answers, of which only one is correct. The Visual Madlibs dataset (Yu et al., 2015) is automatically generated QA pair using "fill in the blank" task.

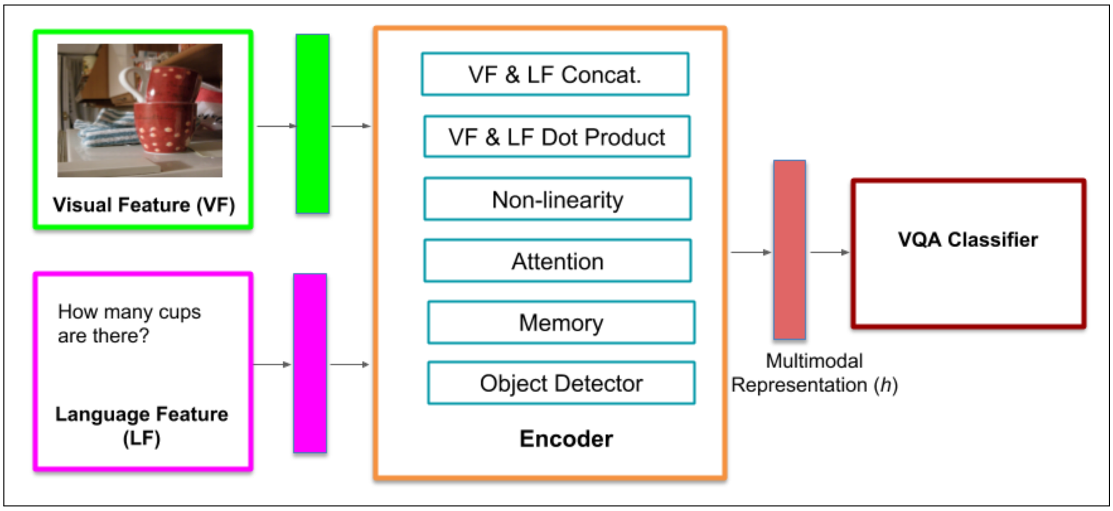


Figure 2.5: A generic VQA Model. A VQA model can have one or more component of the encoder.

Model VQA was presented in Antol et al. (2015). Along with the dataset, different baseline methods are proposed in this work. They model VQA as a classification problem by selecting top-1000 most frequent answers. Their best performing model was based on CNN based feature for images and one hot encoding based on LSTM for questions. The image feature is mapped into the LSTM feature space using a linear transform. Then, these two features are combined using the element-wise multiplication. This combined feature is used to train multilayer perceptron for classification. Soft-max is performed over the output layer to get the classification output, more details in Section 4.3.1.

Similar to Antol et al. (2015), Ren et al. (2015a) also used CNN and LSTM feature, main difference is in terms of how the CNN feature is combined with the LSTM feature Ren et al. (2015a) used the CNN feature also as the first input to the LSTM network followed by vector encoding of each words in the sentence followed by the CNN feature again as the last input. A slight variant of this approach is used in Malinowski et al. (2015), which uses the CNN feature at every step of the LSTM input. They concatenated the CNN feature with each words vector encoding and provided these concatenated features as input to the LSTM network. Gao et al. (2015) models different LSTM networks for the question and the answer by sharing the word embedding layer. The CNN feature is fused with the output of the LSTM. This model can answer more than one-word answers to questions, along with a coherent sentence.

As an alternative to LSTM networks, in Malinowski and Fritz (2014) authors created a Bayesian framework for VQA. They use semantic segmentation to get information about the objects present in an image, such as their categories and spatial locations. Then, their Bayesian framework calculates the probability of each answer given the semantic segmented image features and the question feature. In Kafle and Kanan (2016), they first predict the type of answer like counting, color object, etc. based on the question pre-processing, using Bayesian model, and then it is combined with a discriminative model to provide the final answer. They have used skip-thought vector Kiros et al. (2015) for the question representation.

Park et al. (2016) proposed an interesting idea to create multi-modal space for feature concatenation. Instead of performing element-wise multiplication of the visual and the textual feature, they computed the outer product of two features. To efficiently perform outer products of features, both features are projected into higher dimensional space. Sine outer product is computationally very expensive compared to element-wise multiplication and convolution in Fourier domain is element-wise multiplication in the spatial domain and vice-versa, these high dimensional features are transformed into Furrier domain using Fast Fourier Transform (FFT). In FFT space, element-wise multiplication is performed and then inverse FFT is used to get the feature in the projected higher dimension, which is used for classification purpose. They also used different attention models to improve performance.

In Yang et al. (2015), the author's argue that visual question answering requires multiple steps for reasoning. To provide multiple steps reasoning, they have used multi-layer stacked attention networks. In which, they query image multiple time to provide the answer. Similarly Lu et al. (2016) propose hierarchical co-attention model. They hierarchically put attention on both the question and the image to predict the answer, more details in Section 4.3.1.

2.2 Interactive Tasks

Visually-grounded dialogue has experienced a boost in recent years, in part thanks to the construction of large visual human-human dialogue datasets built by the Computer Vision community (Mostafazadeh et al., 2017; Das et al., 2017a; de Vries et al., 2017). These datasets include two participants, a Questioner, and an Answerer, who ask and answer questions about an image. For example, in the *GuessWhat?!* dataset developed by de Vries et al. (2017), which we exploit in the present work, a Questioner agent needs to guess a target object in a visual scene by asking yes-no questions.

2.2.1 Task-oriented dialogue systems

The conventional architecture of task-oriented dialogue systems includes a pipeline of components, and the task of tracking the dialogue state is typically modeled as a partially-observable Markov decision process (Williams et al., 2013; Young et al., 2013; Kim et al., 2014) that operates on a symbolic dialogue state consisting of predefined variables. The use of symbolic representations to characterize the state of the dialogue has some advantages (e.g., ease of interfacing with knowledge bases), but also some key disadvantages: the variables to be tracked have to be defined in advance and the system needs to be trained on data annotated with explicit state configurations.

Given these limitations, there has been a shift towards neural end-to-end systems that learn their own representations. Early works focus on non-goal-oriented chatbots (Vinyals and Le, 2015; Sordoni et al., 2015b; Serban et al., 2016; Li et al., 2016a,b). Vinyals and Le (2015) proposed an end-to-end system using the sequence to sequence framework. The model takes the previous sentences and predicts the next sentence. The utility of the model is tested on two datasets: a closed-domain IT helpdesk troubleshooting dataset and an open-domain movie transcript dataset.

Similar to Vinyals and Le (2015), Sordoni et al. (2015b) proposed a context sensitive response generation models utilizing the Recurrent Neural Network Language Model (Mikolov et al., 2010). First the past information in a hidden continuous representation, which is then decoded by the RLM to generate contextually relevant plausible responses. Two versions of Dynamic-Context Generative Model is proposed. Serban et al. (2016) further extended the encoding-decoding process by using hierarchical recurrent encoder-decoder(HRED) (Sordoni et al., 2015a) neural network to generate the dialogue. Due to the recurrent hierarchical architecture, the produced dialogue is better. However, in practice tuning, the parameters for the HRED is computationally expensive. Further, improvement in the performance is achieved through the reinforcement learning (Li et al., 2016b) approach.

Bordes et al. (2017) propose a memory network to adopt an end-to-end system to task-oriented dialogue. Recent works combine conventional symbolic with neural approaches Williams et al. (2017); Zhao and Eskenazi (2016); Rastogi et al. (2018), but all focus on language-only dialogue.

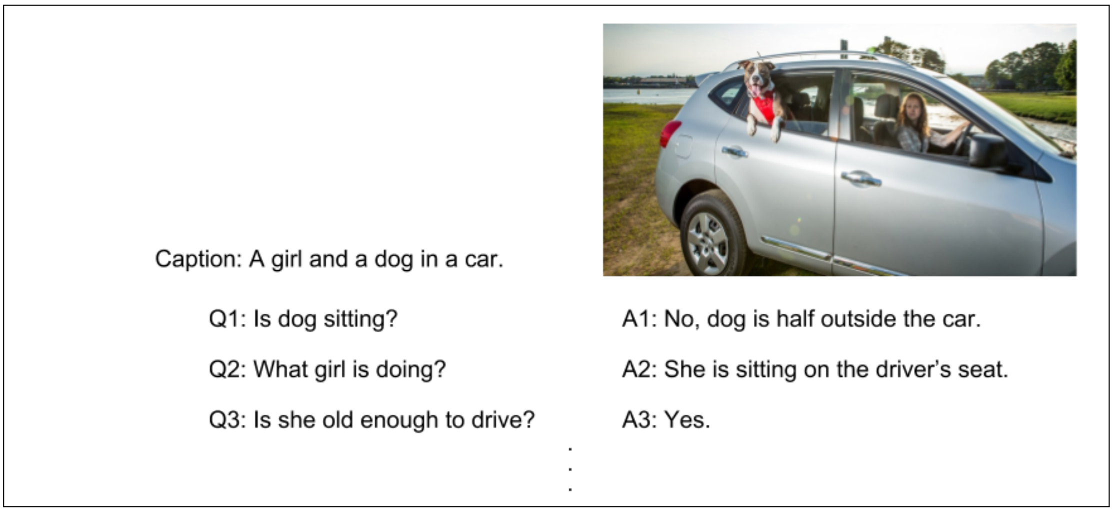
This thesis proposed a visually grounded task-oriented end-to-end dialogue system which, while maintaining the crucial aspect of the interaction of the various modules at play in a conversational agent, grounds them through vision.

Dialogue Manager In traditional dialogue systems, the *dialogue manager* is the core component of a dialogue agent: it integrates the semantic content produced by the interpretation module into the agent's representation of the context (the *dialogue state*) and determines the next action to be performed by the agent, which is transformed into linguistic output by the generation module. Conceptually, a dialogue manager thus includes both (i) a *dialogue state tracker*, which acts as a context model that ideally keeps track of aspects such as current goals, commitments made in the dialogue, entities mentioned, and the level of shared understanding among the participants (Clark, 1996); and (ii) an *action selection policy*, which makes decisions on how to act next, given the current dialogue state. We focus on incorporating a *decision-making module* akin to an action selection policy into a visually-grounded encoder-decoder architecture.

In particular, work on incremental dialogue processing, where a system needs to decide not only *what* to respond but also *when* to act (Rieser and Schlangen, 2011), has some similarities with the problem we address in the present work, namely when to stop asking questions to guess a target. Researchers within the dialogue systems community have applied different approaches to design incremental dialogue policies for how and when to act. Two common approaches are the use of rules parametrized by thresholds that are optimized with human-human data (Buß et al., 2010; Ghigi et al., 2014; Paetzel et al., 2015; Kennington and Schlangen, 2016) and the use of reinforcement learning (Kim et al., 2014; Khouzaimi et al., 2015; Manuvinakurike et al., 2017). For example, Paetzel et al. (2015) implement an agent that aims to identify a target image out of a set of images given descriptive content by its dialogue partner. Decision making is handled by means of a parametrized rule-based policy: the agent keeps waiting for additional descriptive input until either her confidence on a possible referent exceeds a given threshold or a maximum-time threshold is reached (in which case the agent gives up). The thresholds are set up by optimizing points per second on a corpus of human-human dialogues (pairs of participants score a point for each correct guess). In a follow-up paper by Manuvinakurike et al. (2017), the agent's policy is learned with reinforcement learning, achieving higher performance.

We develop a decision-making module that determines, after each question-answer pair in the visually grounded dialogue, whether to ask a further question or to pick a referent in a visual scene. We are interested in investigating the impact of such a module in an architecture that can be trained end-to-end directly from raw data, without specific annotations commonly used in dialogue systems, such as dialogue acts (Paetzel et al., 2015; Manuvinakurike et al., 2017; Kennington and Schlangen, 2016), segment labels (Manuvinakurike et al., 2016), dialogue state features (Williams et al., 2013; Young et al., 2013; Kim et al., 2014), or logical formulas (Yu et al., 2016b).

2.2.2 Visual dialogue



Caption: A girl and a dog in a car.

Q1: Is dog sitting? A1: No, dog is half outside the car.

Q2: What girl is doing? A2: She is sitting on the driver's seat.

Q3: Is she old enough to drive? A3: Yes.

⋮

Figure 2.6: An example Visual Dialogue.

In recent years, researchers in computer vision have proposed tasks that combine visual processing with dialogue interaction. Pertinent datasets created by Das et al. (2017a) and de Vries et al. (2017) include *VisDial* and *GuessWhat?!*, respectively, where two participants ask and answer questions about an image. While impressive progress has been made in combining vision and language, current models make simplifications regarding the integration of these two modalities and their exploitation for task-related actions.

Dataset For visual dialogue, there are two main datasets proposed in the literature: GuessWhat?! and VisDial. The *GuessWhat?!* dataset is a task-oriented dataset and the VisDial dataset is a chit-chat dataset about the image. Both datasets are collected via Amazon Mechanical Turk.

GuessWhat?! Dataset: The *GuessWhat?!* task involves two human participants who see a real-world image, taken from the MS-COCO dataset (Lin et al., 2014a). One of the participants (the Oracle) is assigned a target object in the image and the other participant (the Questioner) has to guess it by asking Yes/No/NA questions to the Oracle. There are no time constraints to play the game. Once the Questioner is ready to make a guess, the list of candidate objects is provided and the game is considered successful if the Questioner picks the target object, more details in Section 3.2.

VisDial Dataset: The VisDial dataset involves two human participants; one participant see the caption corresponding to that image (called 'Questioner'). and another participant sees both image and caption (called 'Answerer'). The questioner task is to imagine the image based by asking the questions. The answerer task is to answer the question asked by the questioner. An example game is shown in Figure 2.6. Unlike GuessWhat, answers are multiword answers. In a dialogue, there are exactly 10 rounds of question-answer.

Model Most of the models proposed for visual dialogue is a combination of image captioning and VQA models. For GuessWhat task, multiple models (de Vries et al., 2017; Strub et al., 2017a; Zhu et al., 2017; Lee et al., 2017; Shekhar et al., 2018; Zhang et al., 2018) has been proposed using two disjoint models for the Questioner, called question generator(QGen) and guesser, and the answerer module is called the Oracle. The QGen model based on the sequence to sequence (Sutskever et al., 2014) architecture and the guesser and oracle model is a classification model based on VQA models (more details in Setion 3.2). Further task improvements are proposed using reinforcement learning based approach (Strub et al., 2017b). Lee et al. (2018) proposed an interesting approach to sample questions from the training dataset, however, this approach is not directly comparable to other approaches due to the sampling of the questions.

For the Visdial tasks also models follow the sequence to sequence (Sutskever et al., 2014) architecture. In Das et al. (2017a), authors proposed a supervised learning based approach to the generate the question and answer. Das et al. (2017b) and Chattopadhyay et al. (2017) use reinforcement learning based reward on the guessing image game to improve the performance. Lu et al. (2017) proposed a transfer based learning approach by transferring knowledge from the discriminator task to the generation task. Further, Massiceti et al. (2018) proposed a generative adversarial networks(GAN) (Goodfellow et al., 2014) based model to have diversity in the answer. Gan et al. (2019) proposed a recurrent attention-based model to have multi-step reasoning to improve the performance.

2.3 Diagnostics Dataset and Methodology

Due to the lack of objective evaluation metrics for IC and triviality of VQA task, research groups have started to look closely at the LaVi integration. To overcome the bias uncovered in previous LaVi datasets, community have started proposing the diagnostics datasets which involve distinguishing distractors from a ground-truth text for an image. To understand the characteristics of LaVi encoder, different diagnostics methodologies are proposed to closely look at the encoding of different linguistic and visual properties.

2.3.1 Diagnostics Dataset

Hodosh and Hockenmaier (2016) shows that contrarily to what prior research had suggested, the image captioning (IC) task is far from been solved. They proposed a series of binary forced-choice tasks such that each task focus on a different aspect of the captions. They evaluate a number of state-of-the-art LaVi algorithms in the presence of distractors and show that IC models are not able to distinguish between a correct and distractor caption. Their evaluation was however limited to a small dataset (namely, Flickr30K (Young et al., 2014)) and the caption generation was based on a hand-crafted scheme using only inter-dataset distractors and the task was comparatively simple, where the model has to select the correct caption for a given image provided with both correct and distractor caption.

Similarly, for the abstract VQA, Zhang et al. (2016) introduced a binary VQA task along with a dataset composed of sets of similar artificial images. They created the dataset such that language biased are controlled, and to perform the task, visual information is essential for the model. This allows for more precise diagnostics of a system's errors.

Goyal et al. (2016a) found that there is a language bias in the VQAv1 (Antol et al., 2015) dataset. The bias is exploited by DNN based model to perform the task. To reduce language bias, they have proposed to balance the dataset in the visual space by collecting complementary images for each question in the VQAv1 dataset, such that the answer to the question for the new image is different. This reduces the language bias and makes harder for the model to exploit the language biases. However, collecting the complementary image is a very expensive process and creates the need for an automatic process to balance the dataset.

Agrawal et al. (2018) created an automatic bias sensitive dataset split for the VQA task. The split, named VQA Under Changing Priors (VQA-CP), is created such that there is a large difference in the answer distribution of train and test set. However, the VQA-CP split mostly checks that memorization capability of the model. Similarly, Lu et al. (2018) introduce a robust captioning split of the COCO captioning dataset (Lin et al., 2014a), using the co-occurrence statistics for COCO object categories. These object categories are used such that the distribution of co-occurring objects differs significantly between training and test.

Similar to the FOIL dataset, Ding et al. (2016) propose to extend the MS-COCO dataset by generating decoys from human-created image captions. They also suggest an evaluation similar to our T1, requiring the LaVi system to detect the true target caption amongst the decoys. Our efforts, however, differ in some substantial ways. First, their technique to create incorrect captions (using BLEU to set an upper similarity threshold) is so that many of those captions will differ from the gold description in more than one respect. For instance, the caption *two elephants standing next to each other in a grass field* is associated with the decoy *a herd of giraffes standing next to each other in a dirt field* (errors: *herd*, *giraffe*, *dirt*) or with *animals are gathering next to each other in a dirt field* (error: *dirt*; infelicities: *animals* and *gathering*, which are both pragmatically odd). Clearly, the more the caption changes in the decoy, the easier the task becomes. In contrast, the foil captions we propose only differ from the gold description by *one* word and are thus more challenging. Secondly, the automatic caption generation of Ding et al means that 'correct' descriptions can be produced, resulting in some confusion in human responses to the task. We made sure to prevent such cases, and human performance on our dataset is thus close to 100%. We note as well that our task does not require any complex instructions for the annotation, indicating that it is intuitive to human beings. Thirdly, their evaluation is a multiple-choice task, where the system has to compare all captions to understand which one is *closest* to the image. This is arguably a simpler task than the one we propose, where a caption is given and the system is asked to classify it as correct or foil: detecting a *correct* caption is much easier than detecting foils. So evaluating precision on both gold and foil items is crucial.

Recently, Hu et al. (2019) proposed a Binary Image Selection (BISON) dataset which propose a decoy image for a given caption. In principle, the dataset is similar to the FOIL dataset, instead of the decoy caption they have proposed the decoy caption. However, the dataset is collected using the crowdsourcing and finding the decoy image is costly, while the FOIL dataset is automatically generated.

For artificial images Johnson et al. (2017) proposed the CLEVR dataset for the diagnostic evaluation of VQA systems. This dataset was designed with the explicit goal of enabling detailed analysis of different aspects of visual reasoning, by minimizing dataset biases and providing rich ground-truth representations for both images and questions. Suhr et al. (2017) proposed an artificial image dataset to test different linguistic phenomena requiring both visual and set-theoretic reasoning.

2.3.2 Diagnostics Methodology

Our work is part of a recent research trend that aims at analyzing, interpreting, and evaluating neural models by means of auxiliary tasks besides the task they have been trained for (Adi et al., 2017; Linzen et al., 2016; Alishahi et al., 2017; Zhang and Bowman, 2018; Conneau et al., 2018). Adi et al. (2017) propose a methodology that facilitates comparing sentence embeddings on a finer-grained level. Specifically, they focused on the length of sentence, the presence of a word in the sentence, and the order of word in the sentence. Linzen et al. (2016) evaluated neural network architectures sentence representation based on the different grammatical complexity focusing on the subject-verb agreement. They found out that given explicit supervision LSTMs could learn to approximate structure-sensitive dependencies. Alishahi et al. (2017) analyzed the representation and encoding of phonemes in RNN using the grounded speech signal. They have shown that phoneme information is saliently present in the lower layer of RNN. Zhang and Bowman (2018) studied the effect of pre-training task on the type of linguistic knowledge is learn by the model. They found out that the language model learned using bidirectional language models do better compare to the translation model in extracting syntax information. They also show that randomly initialized model performs reasonably well. Conneau et al. (2018) introduce ten probing tasks to infer the type of information is stored in the sentence embedding vector. They show that the sentence embeddings are capturing a wide range of linguistic knowledge.

Chapter 3

Jointly Learning to See, Ask, and GuessWhat

In this chapter, a grounded dialogue state encoder is proposed which addresses a foundational issue on how to integrate visual grounding with dialogue system components. The proposed visually-grounded encoder leverages synergies between guessing and asking questions, as it is trained jointly using multi-task learning. The model is further enriched via cooperative learning. We show that the introduction of both the joint architecture and cooperative learning lead to accuracy improvements over the baseline system and provide in-depth analysis to show that the linguistic skills of the models differ dramatically, despite approaching comparable performance levels. This points at the importance of analyzing the linguistic output of competing systems beyond numeric comparison solely based on task success.[1]

[1] Part of work of this chapter will appear in NAACL 2019 as
Ravi Shekhar, Aashish Venkatesh, Tim Baumgärtner, Elia Bruni, Barbara Plank, Raffaella Bernardi, and Raquel Fernández, "**Beyond task success: A closer look at jointly learning to see, ask, and GuessWhat** ", *In Proc. of* 17^{th} *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) 2019 (Long-Oral).*

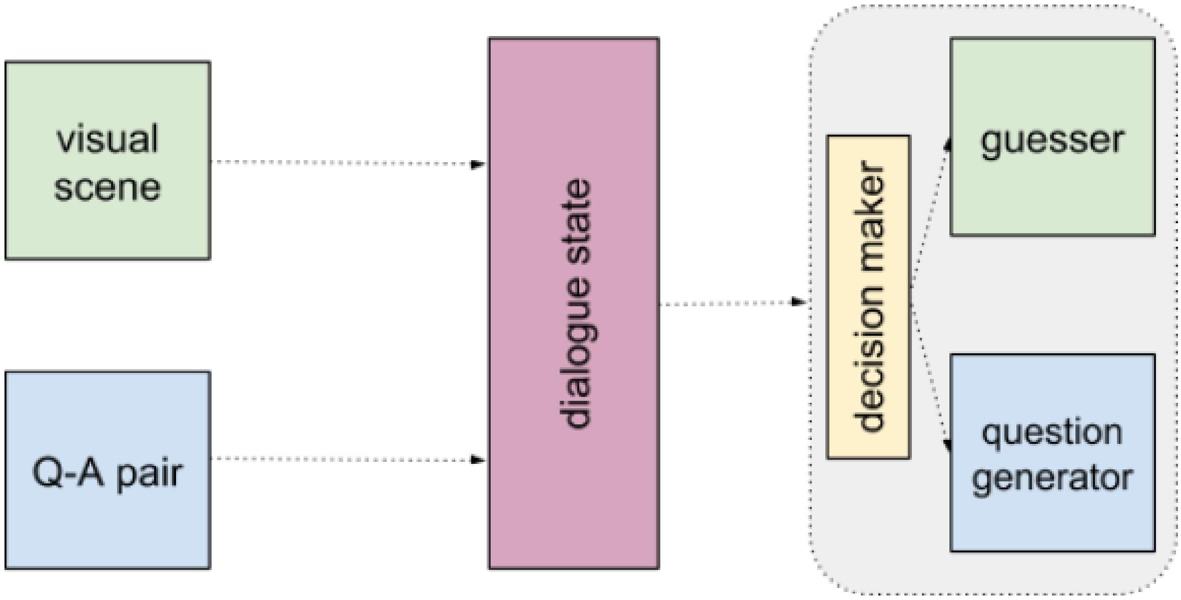


Figure 3.1: Our proposed Questioner model with a decision making component.

3.1 Introduction

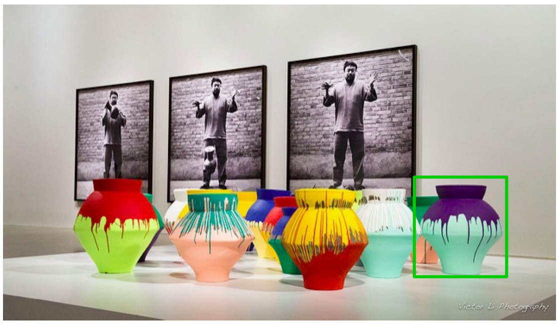
Over the last few decades, substantial progress has been made in developing dialogue systems that address the abilities that need to be put to work during conversations: Understanding and generating natural language, planning actions, and tracking the information exchanged by the dialogue participants. The latter is particularly critical since, for communication to be effective, participants need to represent the state of the dialogue and the common ground established through the conversation (Stalnaker, 1978; Lewis, 1979; Clark, 1996).

In addition to the challenges above, the dialogue is often situated in a perceptual environment. In this study, we develop a dialogue agent that builds a representation of the context and the dialogue state by integrating information from both the *visual* and *linguistic* modalities. We take the *GuessWhat?!* game de Vries et al. (2017) as our test-bed and model the agent in the Questioner's role.

To model the Questioner, previous work relies on two independent models to learn to ask questions and to guess the target object, each equipped with its own encoder (de Vries et al., 2017; Strub et al., 2017a; Zhu et al., 2017; Lee et al., 2017; Shekhar et al., 2018; Zhang et al., 2018). In contrast, we propose an end-to-end architecture with a single *visually-grounded dialogue state encoder* with a decision making(DM) component (Figure 3.1). As shown in Figure 3.1, both visual and textual (QA-pair) is first merged to form a dialogue state (details in Section 3.3). Using this dialogue state, the DM makes the decision to 'ask', further question using question generator, or 'guess', the target object using the Guesser. Our system is trained jointly in a supervised learning setup, extended with cooperative learning (CL) regime: By letting the model play the game with self-generated dialogues, the components of the Questioner agent learn to better perform the overall Questioner's task in a cooperative manner. Das et al. (2017b) have explored the use of CL to train two visual dialogue agents that receive joint rewards when they play a game successfully. To our knowledge, ours is the first approach where cooperative learning is applied to the internal components of a grounded conversational agent.

3.2 GuessWhat?!

The *GuessWhat?!* game is two player task oriented visual dialogue game. It is a game between two agents and the goal is to locate an object in an image by asking a series of questions. The *GuessWhat?!* game requires the agent to ask questions about the image to narrow down the possible target object. This requires the agent to have spatial and effective language understanding. An example of the game is shown in Figure 3.2.



QUESTIONER	**ORACLE**
Is it a vase?	*Yes*
Is it on the left of the picture?	*No*
Is it between two vases?	*No*
Is it the light blue one?	*Yes*
Is it on the edge of the table?	*Yes*

Figure 3.2: An example game.The Questioner asks questions to the Oracle to locate the object marked by green bounding box. Source:de Vries et al. (2017)

GuessWhat?! Dataset

GuessWhat?! is a cooperative game between two agents where both the agents have access to the same image. The task involves two human participants who see a real-world image, taken from the MS-COCO dataset (Lin et al., 2014a). The objective of the game is for the Questioner to have a dialogue with the Oracle to find the required object in the image. The Oracle is randomly assigned an object in the image and has to answer the questions about the object with Yes, No or Not Applicable (NA). The Questioner does not know about the assigned object to the Oracle and has to ask questions to gain more information about the object. When the Questioner thinks it has sufficient information to guess the object, it is presented with a list of candidate objects (max 20) from which it has to choose the correct object given the image, and the dialogue history.

The dataset is collected using AMT. The dataset consists of 66,537 unique images with 155,280 games/dialogues with an average of 2.3 games per image. There are 821,889 question-answer (QA) pairs amongst these games with a mean of 5.2 QA pairs per game. The answers in the QA pairs have the distribution: 52.2% *No*, 45.6% *Yes* and 2.2% *NA*. In the collected dataset de Vries et al. (2017), about 84.6% of the games are successful, 8.4% unsuccessful and 7% of the games are incomplete. The GW dataset is split into training, validation and test split by randomly allocating 46794, 9844 and 9899 unique images, respectively. Table 3.1 provides the statistics of the dataset. We use the official splits given by the authors at guesswhat.ai for all the experiments.

	# Unique Images	# Dialogues	# QA-pairs
train	46794	113221	579633
val	9844	23739	120318
test	9899	23785	121938

Table 3.1: GuessWhat?! Dataset statistics.

GuessWhat?! Baseline Model

Initial models, proposed by de Vries et al. (2017), use supervised learning (SL): the Questioner and the Answerer are trained to generate utterances (by word sampling) that are similar to the human gold standard.

Oracle The task of the Oracle is similar to VQA. The Oracle has to answer Yes/No/NA given a question or the dialogue history and the assigned object information such as bounding-box, category, image crop, and image. The best performing model variant of the baseline uses the present question with the object's bounding box and category information. As shown in Figure 3.3, the Oracle comprises of an LSTM and an MLP. The question, q_t, from the Questioner is processed by the LSTM (LSTM$_o$). The input to the Oracle is category embedding, c, and handcrafted spatial features, $x_{spatial}$, extracted from the bounding box which is concatenated with the last hidden state of the LSTM$_o$, hs$_o$. The

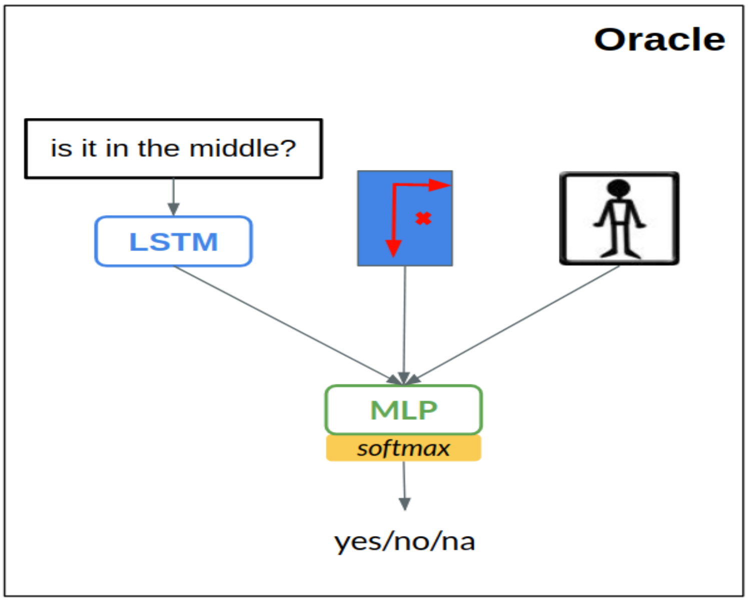


Figure 3.3: **Baseline Oracle Model**

hand-crafted spatial features, proposed by Hu et al. (2016), are constructed from the bounding box information to form an 8-dimensional vector as shown below:

$$x_{spatial} = [x_{min}, y_{min}, x_{max}, y_{max}, x_{center}, y_{center}, w_{box}, h_{box}] \tag{3.1}$$

where w_{box} and h_{box} are the width and height of the bounding box, respectively. The image height and width are normalised to be between 1 to -1 and the origin is placed at the center of the image. The MLP (MLP$_o$) processes the input concatenated features to produce an answer Yes/No/NA.

Questioner The Questioner's objective is to ask relevant questions about the image to locate the object. After having accumulated evidence to locate the object, it is presented with a list of candidate objects from which it has to pick target one. So, the questioner has two tasks: to ask questions and the action of choosing the object. This is modeled by de Vries et al. (2017) using two independent models which are called the Question Generator and the Guesser. These models are explained below.

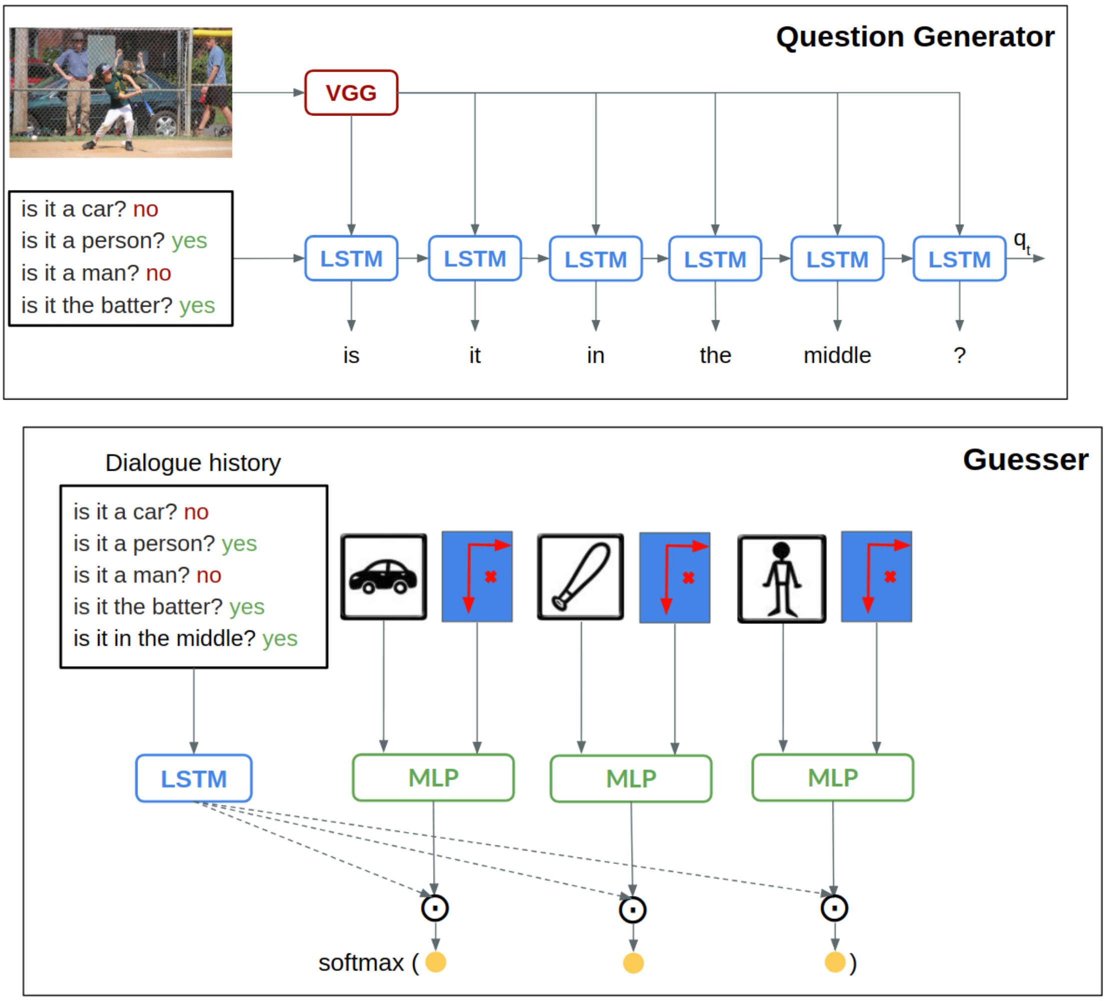


Figure 3.4: **Baseline Questioner Model:** Questioner Model: Question Generator (up) and Guesser (down)

Question Generator(QGen) QGen is implemented as a Recurrent Neural Network (RNN) with a transition function handled with Long-Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), on which a probabilistic sequence model is built with a Softmax classifier. At each time step in the dialogue, the model receives as input the raw image and the dialogue history and generates the next question one word at a time. The image is encoded by extracting its VGG-16 features (Simonyan and Zisserman, 2014).

Guesser The Guesser model exploits the annotations in the MS-COCO dataset (Lin et al., 2014a) to represent candidate objects by their object category and their spatial coordinates. This yields better performance than using raw image features in this case, as reported by de Vries et al. (2017). The objects' categories and coordinates are passed through a Multi-Layer Perceptron (MLP) to get an embedding for each object. The Guesser also takes as input the dialogue history processed by its own dedicated LSTM. A dot product between the hidden state of the LSTM and each of the object embeddings returns a score for each candidate object.

Limitations In this chapter, we will address the following two major limitations of the baseline model proposed in de Vries et al. (2017):

- The Questioner agent is composed of two independent models, QGen and Guesser. Due to this, the two modules are detached from the context of each other.

- The Questioner has to always ask a fixed number of questions before being able to guess the target object. This causes the QGen to ask questions even if it has enough evidence to guess and leads to redundant questions. Also, if the maximum number of questions is set very low, the Guesser will not be able to locate the target object.

Apart from the above two limitations, one major drawback of current setting is that the Guesser model have to be provided metadata, i.e., spatial co-ordinate, category, of the objects in the image to guess the target object. This could be addressed using the object detector, but this issue is outside the scope of this thesis.

3.3 Visually-grounded dialogue state encoder

In line with the baseline model, our Questioner agent includes two submodules, a QGen and a Guesser. As in the baseline, the Guesser guesses after a fixed number of questions, which is a parameter tuned on the validation set. Our agent architecture differs from the baseline model by de Vries et al. (2017): Rather than operating independently, the language generation and guessing modules are connected through a common *grounded dialogue state encoder* (GDSE) which combines linguistic and visual information as a prior for the two modules. Given this representation, we will refer to our Questioner agent as GDSE.

As illustrated in Figure 3.5, the encoder receives as input representations of the visual and linguistic context. The visual representation consists of the second to last layer of ResNet152 trained on ImageNet. The linguistic representation is obtained by an LSTM (LSTM$_e$) which processes each new question-answer pair in the dialogue. At each question-answer QA_t, the last hidden state of LSTM$_e$ is concatenated with the image features I, passed through a linear layer and a *tanh* activation to result in the final layer h_t:

$$h_t = \tanh\left(W \cdot [\text{LSTM}_e(qa_{1:t-1});\ I]\right) \tag{3.2}$$

where $[\cdot;\cdot]$ represents concatenation, $I \in \mathbb{R}^{2048\times1}$, LSTM$_e$ $\in \mathbb{R}^{1024\times1}$ and $W \in \mathbb{R}^{512\times3072}$ (identical to prior work except for tuning the ResNet-specific parameters). We refer to this final layer as the *dialogue state*, which is given as input to both QGen and Guesser.

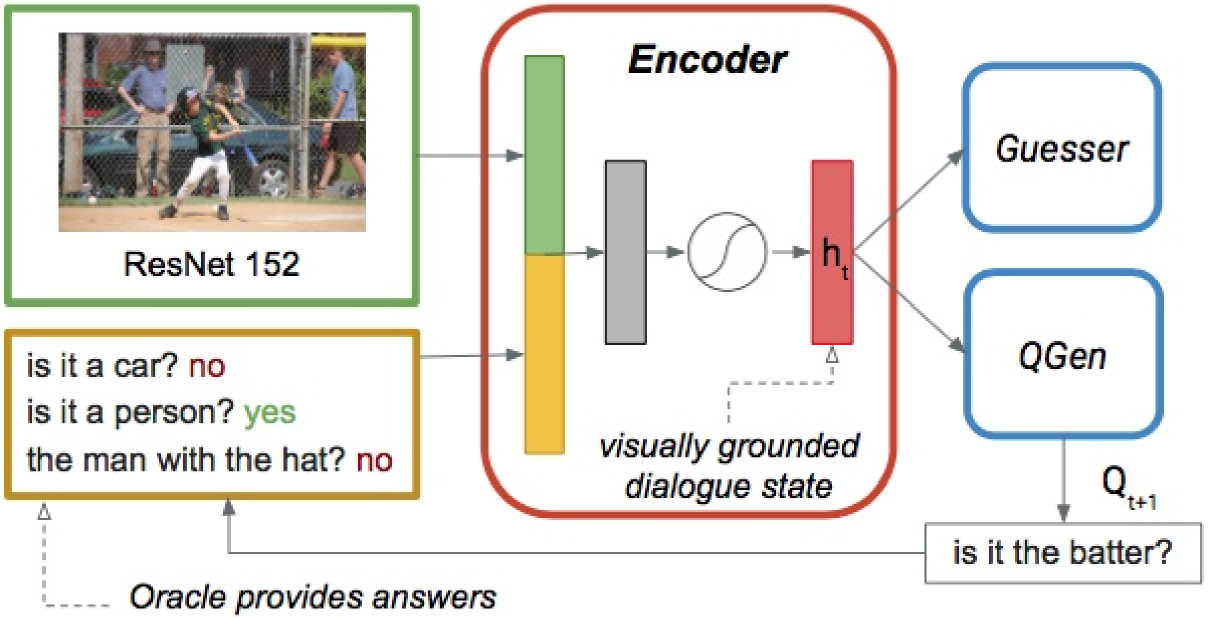


Figure 3.5: Our questioner model with a single visually grounded dialogue state encoder.

As illustrated in Figure 3.6, our QGen, and Guesser modules are like the corresponding modules by de Vries et al. (2017), except for the crucial fact that they receive as input the same grounded dialogue state representation. QGen employs an LSTM (LSTM$_q$) to generate the token sequence for each question conditioned on h_t, which is used to initialise the hidden state of LSTM$_q$. As input at every time step, QGen receives a dense embedding of the previously generated token w_{i-1} and the image features I:

$$p(w_i) = p(w_i|w_1, ..., w_{i-1}, h_t, I) \tag{3.3}$$

We optimise QGen by minimising the Negative Log Likelihood (NLL) of the human dialogues and use the Adam optimiser (Kingma and Ba, 2014a):

$$\mathcal{L}_Q = \sum_i - \log p(w_i) \tag{3.4}$$

Thus, in our architecture the LSTM$_q$ of QGen in combination with the LSTM$_e$ of the Encoder form a sequence-to-sequence model (Sutskever et al., 2014), conditioned on the visual and linguistic context — in contrast to the baseline

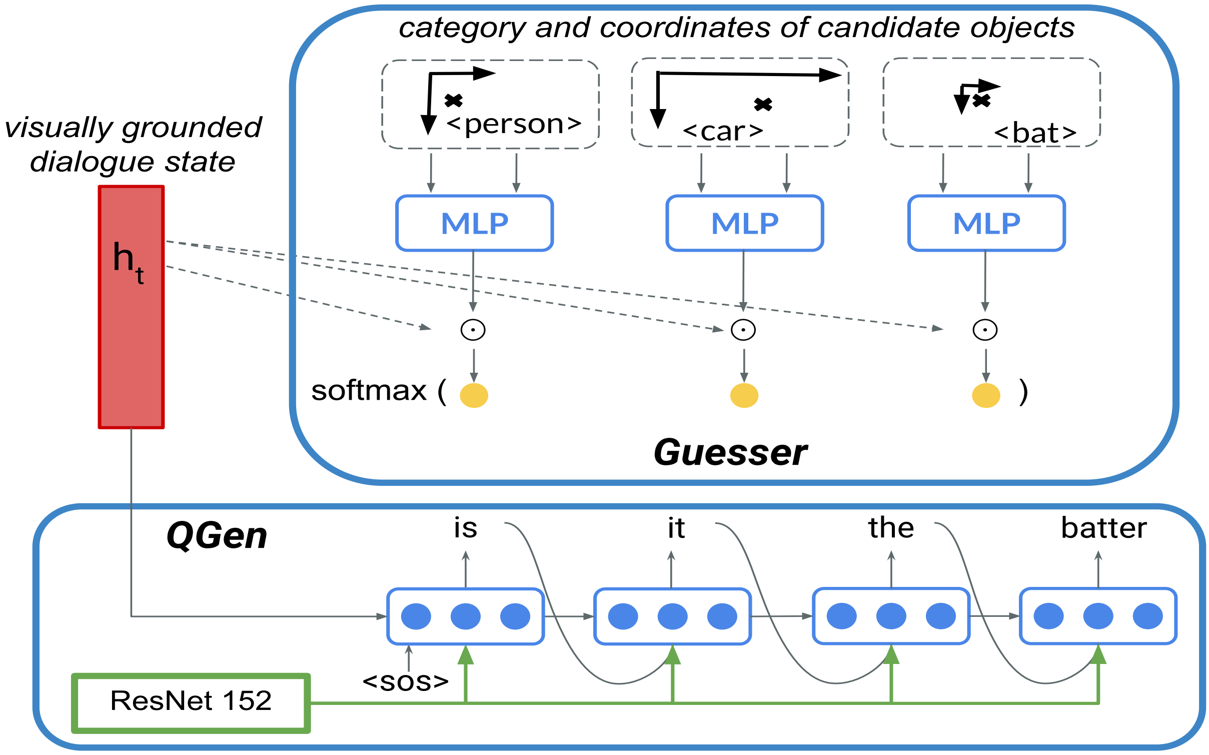


Figure 3.6: Question Generation and Guesser modules.

model, where question generation is performed by a single LSTM on its own.

The Guesser consists of an MLP which is evaluated for each candidate object in the image. It takes the dense embedding of the category and the spatial information of the object to establish a representation $r_j \in \mathbb{R}^{512\times 1}$ for each object. A score is calculated for each object by performing the dot product between the dialogue state h_t and the object representation. Finally, a softmax over the scores results in a probability distribution over the candidate objects:

$$p(o_j) = \frac{e^{h_t^T \cdot r_j}}{\sum_j e^{h_t^T \cdot r_j}} \tag{3.5}$$

We pick the object with the highest probability and the game is successful if $o_{guess} = o_{target}$, where $o_{guess} = \arg\max_j p(o_j)$. As with QGen, we optimise the Guesser by minimising the NLL and again make use of Adam:

$$\mathcal{L}_G = -\log p(o_{target}) \tag{3.6}$$

The resulting architecture is fully differentiable. In addition, the GDSE agent faces a multi-task optimization problem: While the QGen optimizes $\mathcal{L}_Q$ and the Guesser optimizes $\mathcal{L}_G$, the parameters of the Encoder (W, LSTM$_e$) are optimized via both $\mathcal{L}_Q$ and $\mathcal{L}_G$. Hence, both tasks faced by the Questioner agent contribute to the optimization of the dialogue state h_t, and thus to more efficient encoding of the input context.

Decision Making We further extend the GDSE with a decision-making component (DM) (cf. Figure 3.1) that determines, after each question/answer pair, whether QGen should *ask* another question or whether the Guesser should *guess* the target object. We treat this decision problem as a binary classification task, for which we use an MLP followed by a Softmax function that outputs probabilities for the two classes of interest: *ask* and *guess*. The argmax function then determines the class of the next action. With this approach, we bypass the need to specify any decision thresholds and instead let the model learn whether enough evidence has been accumulated during the dialogue so far to let the Guesser pick up a referent.

3.4 Learning Approach

We first introduce the supervised learning approach used to train both BL and GDSE, then our cooperative learning regime, and finally the reinforcement learning approach we compare to.

3.4.1 Supervised learning

In the baseline model, the QGen and the Guesser modules are trained autonomously with supervised learning (SL): QGen is trained to replicate human questions and, independently, the Guesser is trained to predict the target object.

Our new architecture with a common dialogue state encoder allows us to formulate these two tasks as a multi-task problem, with two different losses (Eq. 3.4 and 3.6 in Section 3.3).

These two tasks are not equally difficult: While the Guesser has to learn the probability distribution of the set of possible objects in the image, QGen needs to fit the distribution of natural language words. Thus, QGen has a harder task to optimize and requires more parameters and training iterations. We address this issue by making the learning schedule task-dependent. We call this setup *modulo-n* training, where *n* indicates after how many epochs of QGen training the Guesser is updated together with QGen.

As there are no labels for the DM about when to ask more questions and when to guess the object, we follow the label generation procedure introduced by Shekhar et al. (2018). The labels for the Decider are generated by annotating all the last question-answer pairs in the games with *guess* and other question-answer pairs as *ask*. So, we have an unbalanced dataset for the decider module where the guess label makes up for only 20%. We address this class imbalance by adding a weighting factor, α, to the loss. The balanced loss is given by

$$\mathcal{L}_D = \alpha_{target} \cdot (-\log p(\mathrm{dec}_{target})) \tag{3.7}$$

where $\alpha_{guess} = 0.8$ and $\alpha_{ask} = 0.2$. During inference, we continue to ask questions unless the Decider chooses to end the conversation or the maximum number of questions has been reached. The architecture of the Decider consists only of the MLP_d.

Using the validation set, we experimented with *n* from 5 to 15 and found that updating the Guesser every 7 epochs worked best. With this optimal configuration, we then train GDSE for 100 epochs (batch size of 1024, Adam, learning rate of 0.0001) and select the Questioner module best performing on the validation set (henceforth, GDSE-SL or simply SL, and with Decision making GDSE-SL-DM or SL-DM).

3.4.2 Cooperative learning

Once the model has been trained with SL, new training data can be generated by letting the agent play new games. Given an image from the training set used in the SL phase, we generate a new training instance by randomly sampling a target object from all objects in the image. We then let our Questioner agent and the Oracle play the game with that object as a target, and further train the common encoder using the generated dialogues by backpropagating the error with gradient descent through the Guesser. After training the Guesser and the encoder with generated dialogues, QGen needs to 're-adapt' to the newly arranged encoder parameters. To achieve this, we re-train QGen on the human data with SL, but using the new encoder states. Also here, the error is back-propagated with gradient descent through the common encoder. To update the parameters of the DM, we use the guesser output as the label to decide 'ask' or 'guess'. When the guesser is successful, the DM has to 'guess' else it has to 'ask' further question.

Regarding *modulo-n*, in this case QGen is updated at every n^{th} epoch, while the Guesser is updated at all other epochs; we experimented with *n* from 3-7 and set it to the optimal value of 5.

The GDSE previously trained with SL is further trained with this cooperative learning regime for 100 epochs (batch size of 256, Adam, learning rate of 0.0001), and we select the Questioner module performing best on the validation set (henceforth, GDSE-CL or simply CL and with Decision making GDSE-CL-DM or CL-DM).

3.4.3 Reinforcement learning

Strub et al. (2017a) proposed the first extension of BL de Vries et al. (2017) with deep reinforcement learning (RL). They present an architecture for end-to-end training using an RL policy. First, the Oracle, Guesser, and QGen models are trained independently using supervised learning. Then, QGen is further trained using a policy gradient.

We use the publicly available code and pre-trained model based on Sampling Strub et al. (2017a), which resulted in the closest performance to what was reported. by the authors.[2] This is the RL model we use throughout the rest of the chapter.

3.5 Experiments and Results

We use the same train (70%), validation (15%), and test (15%) splits as de Vries et al. (2017). The test set contains new images not seen during training. We use two experimental setups for the number of questions to be asked by the question generator, motivated by prior work: 5 questions (5Q) following de Vries et al. (2017), and 8 questions (8Q) as in Strub et al. (2017a).

For evaluation, we report task success in terms of accuracy Strub et al. (2017a). To neutralize the effect of random sampling in training CL, we trained the model 3 times. RL is tested 3 times with sampling. We report means and standard deviation (for some tables these are provided in the supplementary material).

Table 5.2 reports the results for all models. There are several take-aways.

Grounded joint architecture First of all, our visually-grounded dialogue state encoder is effective. GDSE-SL outperforms the baseline by de Vries et al. (2017) significantly in both setups (absolute accuracy improvements of 6.6% and 9%). To evaluate the impact of the multi-task learning aspect, we did an ablation

[2] Their result of 53.3% accuracy published in Strub et al. (2017a) is obsolete, as stated on their GitHub page (`https://github.com/GuessWhatGame/guesswhat`) where they report 56.5% for sampling and 58.4% for greedy search. By running their code, we could only replicate their results with sampling, obtaining 56%, while greedy and beam search resulted in similar or worse performance. Our analysis showed that greedy and beam search have the additional disadvantage of learning a smaller vocabulary.

Model	5Q	8Q
Baseline	41.2	40.7
GDSE-SL	47.8	49.7
GDSE-CL	53.7 (±.83)	58.4 (±.12)
RL	56.2 (±.24)	56.3 (±.05)
GDSE-SL-DM	46.78	49.12
GDSE-CL-DM	4 9.77(±1.16)	53.89(±.24)

Table 3.2: Test set accuracy for each model (for setups with 5 and 8 questions). GDSE-SL is our grounded supervised learning system, GDSE-CL the cooperative learning setup, and RL the results we obtain with the reinforcement learning system by Strub et al. (2017). The average number of question asked by DM models for 5Q setting is 3.83 and 4.02(±0.10) for SL-DM and CL-DM respectively. Moreover, in the case of 8Q setting, the average number of question is 5.49 and 5.46(±0.10) for SL-DM and CL-DM respectively.

study and used the encoder-decoder architecture to train the QGen and Guesser modules independently. With such a decoupled training we obtain lower results: 44% and 43.7% accuracy for 5Q and 8Q, respectively. Hence, the multi-task component brings an increase of up to 6% over the baseline.[3]

Cooperative learning and RL The introduction of the cooperative learning approach results in a clear improvement over GDSE-SL: +8.7% (8Q: from 49.7 to 58.4) and +5.9% (with 5Q). Despite its simplicity, our GDSE-CL model achieves a task success rate which is comparable to RL: In the 8Q setup, GDSE-CL reaches an average accuracy of 58.4 versus 56.3 for RL, giving CL a slight edge in this setup (+2.1%), while in the 5Q setup RL is slightly better (+2.5%). Overall, the accuracy of the CL and RL models is close. The interesting question is how the linguistic skills and strategy of these two models differ, to which we turn in the next section.

We compared to Strub et al. (2017a), but RL has also been put forward

[3] While de Vries et al. (2017) originally report an accuracy of 46.8%, this result was later revised to 40.8%, as clarified on their GitHub page. Our own implementation of the baseline system achieves an accuracy of 41.2%.

by Zhang et al. (2018), who report 60.7% accuracy (5Q). This result is close to our highest GDSE-CL result (60.8 ±0.51, when optimized for 10Q).[4] Their RL system integrates several partial reward functions to increase coherence, which is an interesting aspect. Yet their code is not publicly available. We leave the comparison to Zhang et al. (2018) and adding RL to GDSE to future work.

Utility of DM The lower part of Table 5.2 reports the accuracy using the DM with both SL and CL. We can see that the overall accuracy is decreasing in the case of DM, for SL by $0.5 - 1\%$ and for CL by $4 - 5\%$. However, the model asks comparatively very fewer questions and behave more similarly to humans. For more than the 50% of games, the DM decides to stop asking questions before reaching the max number of questions allowed. In the case of 8Q, it is asking only 5.49 and $5.46(\pm0.10)$ questions for SL-DM and CL-DM respectively, which is very low. The significant drop in accuracy for CL could be due to the signal the model is getting while training. Since CL model is being trained on the generated data, sometimes it might produce the wrong signal.

3.6 Analysis using GDSE

In this section, we present a range of analyses that aim to shed light on the performance of the models. They are carried out on the test set data using the 8Q setting, which yields better results than the 5Q setting for the GDSE models and RL. Given that there is only a small difference in accuracy for the baseline with 5Q and 8Q, for comparability, we analyze dialogues with 8Q also for BL.

[4] Since our aim is to compare to the best setup for BL (5Q) and RL (8Q), we do not report our results with 10Q in Table 5.2.

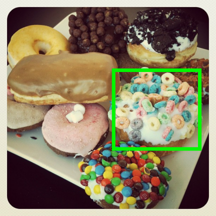
3.6.1 Quantitative analysis of linguistic output

We analyze the language produced by the Questioner agent with respect to three factors: (1) lexical diversity, measured as type/token ratio over all games, (2) question diversity, measured as the percentage of unique questions over all games, and (3) the number of games with questions repeated verbatim. We compute these factors on the test set for the models and for the human data (H).

As shown in Table 3.3, the linguistic output of SL & CL is closer to the language used by humans: Our agent is able to produce a much richer and less repetitive output than both BL and RL. In particular, it learns to use a more diverse vocabulary, generates more unique questions, and repeats questions within the same dialogue at a much lower rate than the baseline and RL: 93.5% of the games played by BL contain at least one verbatim question repetition, for RL this happens in 96.47% of the cases, whereas for SL and CL this is for only 55.8% and 52.19% of the games, respectively. Notice the % Games with repeated Questions in case of the DM model, there is a large drop in the repetition over GDSE-SL: -13.33% (from 55.80 to 42.47) and GDSE-CL: -19.68% (from 52.19 to 32.51). Significant drop in the accuracy also points to the strong case for having a DM.

	Lexical diversity	Question diversity	% Games with repeated Q's
BL	0.030	1.60	93.50
SL	0.101	13.61	55.80
CL	0.115 (±.02)	14.15 (±3.0)	52.19 (±4.7)
RL	0.073 (±.00)	1.04 (±.03)	96.47 (±.04)
SL-DM	0.047	1.62	42.47
CL-DM	0.135(±.02)	10.25(±2.46)	32.51(±6.45)
H	0.731	47.89	—

Table 3.3: Statistics of the linguistic output of all models with the 8Q setting and of humans (H) in all test games.



Humans	*[success]*	**GDSE-CL**	*[success]*	**RL**	*[failure]*
1. does it have cereal on it?	no	1. is it a donut?	yes	1. is it food?	yes
2. does it have pink frosting?	no	2. is it on the left?	no	2. is it a donut?	yes
3. does it have cookies?	no	3. is it on the right?	yes	3. is it in left?	no
4. is it a donut?	yes	4. is it the whole donut?	yes	4. is it in top?	yes
5. does it have maple frosting?	no	5. does it have pink icing?	no	5. is it in top?	yes
6. is there candy?	yes	6. is it touching the donut with the sprinkles?	yes	6. is it in top?	yes
7. is it the doughnut with M&Ms?	no	7. does it have chocolate icing?	no	7. is it in top?	yes
8. is it dark brown?	no	8. is it the third donut?	yes	8. is it top?	yes

Figure 3.7: Game example where GDSE-CL succeeds and RL fails at guessing the target object (green box).

3.6.2 Dialogue strategy

To further understand the variety of questions asked by the agents, we classify questions into different types. We distinguish between questions that aim at getting the category of the target object (ENTITY questions, e.g., *'is it a vehicle?'*) and questions about properties of the queried objects (ATTRIBUTE questions, e.g., *'is it square?'* or *'are they standing?'*). Within ATTRIBUTE questions, we make a distinction between color, shape, size, texture, location, and action questions. Within ENTITY questions, we distinguish questions whose focus is an object category or a super-category. The classification is done by manually extracting keywords for each question type from the human dialogues, and then applying an automatic heuristic that assigns a class to a question given the presence of the relevant keywords.[5] This procedure allows us to classify 91.41% of questions asked by humans. The coverage is higher for the questions asked by the models: 98.88% (BL), 94.72% (SL), 94.11% (CL), 98.30%(SL-DM), 96.57%(CL-DM) and 99.51 % (RL).[6]

The statistics are shown in Table 3.4. We use Kullback-Leibler (KL) divergence to measure how the output of each model differs from the human distribution of fine-grained question classes. The baseline's output has the highest

[5] A question may be tagged with several attribute classes if keywords of different types are present. E.g., *"Is it the white one on the left?"* is classified as both COLOR and LOCATION.

[6] Appendix A provides details on the question classification procedure: the lists of keywords by class, the procedure used to obtain these lists, as well as the pseudo-code of the heuristics used to classify the questions.

degree of divergence: For instance, the BL model does never ask any SHAPE or TEXTURE questions, and hardly any SIZE questions. The output of the RL model also differs substantially from the human dialogues: It asks a very large number of LOCATION questions (74.8% vs. 40% for humans). Our models, in contrast, generate question types that resemble human distribution more closely. However, there is slightly more variation in the case of the DM models. Specifically, SL-DM asks more ENTITY questions (71.03%) compared to other models. We believe this is due to the early stopping of the questions.

Question type	Example	BL	SL	CL	RL	SL-DM	CL-DM	H
ENTITY		**49.00**	**48.07**	**46.51**	**23.99**	**71.03**	**51.36**	**38.11**
SUPER-CAT	*Is it a vehicle?*	19.6	12.38	12.58	14.00	15.35	15.40	14.51
OBJECT	*Is it a skateboard?*	29.4	35.70	33.92	9.99	55.68	35.97	23.61
ATTRIBUTE		**49.88**	**46.64**	**47.60**	**75.52**	**27.27**	**45.21**	**53.29**
COLOR	*Is he wearing blue?*	2.75	13.00	12.51	0.12	10.57	8.41	15.50
SHAPE	*Is it square?*	0.00	0.01	0.02	0.003	0.0	0.07	0.30
SIZE	*The bigger one?*	0.02	0.33	0.39	0.024	0.01	0.67	1.38
TEXTURE	*Is it wood?*	0.00	0.13	0.15	0.013	0.01	0.25	0.89
LOCATION	*The one on the left?*	47.25	37.09	38.54	74.80	21.70	39.92	40.00
ACTION	*Are they standing?*	1.34	7.97	7.60	0.66	3.96	8.01	7.59
Not classified		**1.12**	**5.28**	**5.90**	**0.49**	**1.70**	**3.43**	**8.60**
KL wrt Human distribution		0.953	0.042	0.038	0.396	1.48	0.055	—

Table 3.4: Percentage of questions per question type in all the test set games played by humans (H) and the models with the 8Q setting, and KL divergence from human distribution of fine-grained question types.

We also analyze the structure of the dialogues in terms of the sequences of question types asked. As expected, both humans and models almost always start with an ENTITY question (around 97% for BL, SL and CL, 98.7% for RL, and 78.48% for humans), in particular a SUPER-CATEGORY (around 70% for BL, SL and CL, 84% for RL, and 52.32% for humans). In some cases, humans start by asking questions directly about an attribute that may easily distinguish an object from others, while this is very uncommon for models. Figure 3.7 shows an example: The human dialogue begins with an ATTRIBUTE question (*'does it have cereal on it?'*), which in this case is not very effective and leads to a change in strategy at turn 4. The CL model starts by asking an OBJECT question (*'is it a donut?'*) while the RL model begins with a more generic SUPER-CATEGORY question (*'is it food?'*).

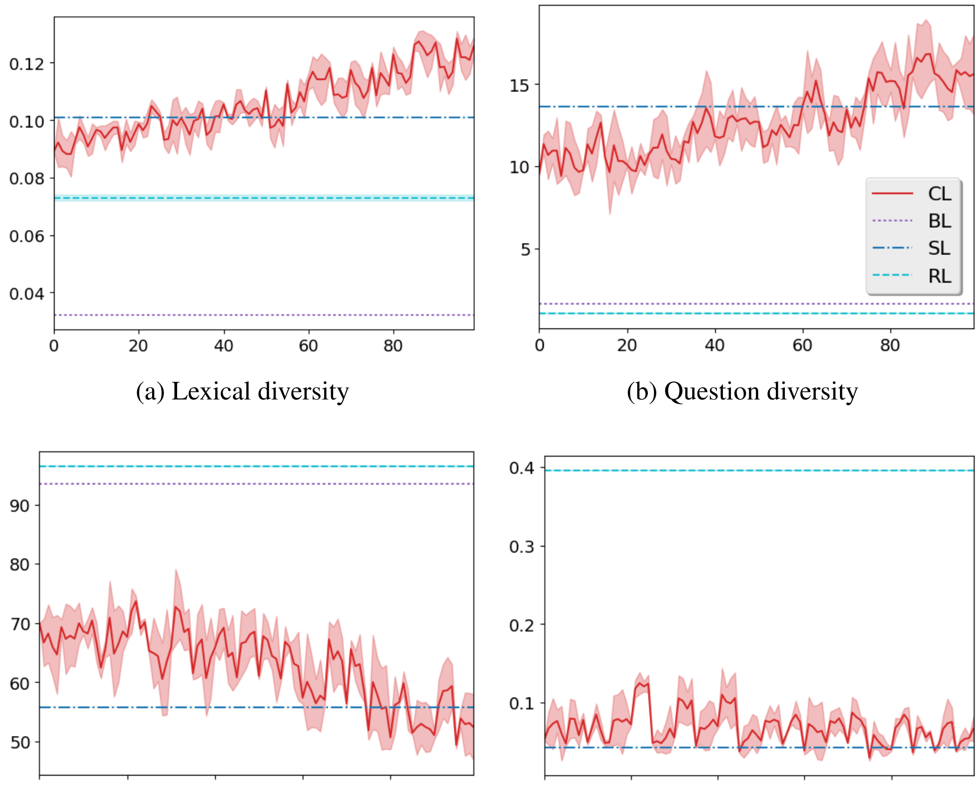
We check how the answer to a given question type affects the type of follow-up question. In principle, we expect to find that question types that are answered positively will be followed by more specific questions. This is indeed what we observe in the human dialogues, as shown in Table 3.5. For example, when a SUPER-CATEGORY question is answered positively, humans follow up with an OBJECT or ATTRIBUTE question 89.56% of the time. This trend is mirrored by all models.

Question type shift	BL	SL	CL	RL	SL-DM	CL-DM	H
SUPER-CAT → OBJ/ATT	89.05	92.61	89.75	95.63	98.65	98.04	89.56
OBJECT → ATTRIBUTE	67.87	60.92	65.06	99.46	91.23	90.51	88.70

Table 3.5: Proportion of question type shift vs. no type shift in consecutive questions $Q_t \rightarrow Q_{t+1}$ where Q_t has received a Yes answer.

Overall, the models also learn the strategy to move from an OBJECT to an ATTRIBUTE question when an OBJECT question receives a Yes answer. The BL, SL, and CL models do this to a lesser extent than humans, while the RL model systematically transitions to attributes (in 99.46% of cases), using mostly LOCATION questions as pointed out above. For example (Figure 3.7), after receiving an affirmative answer to the OBJECT question 'is it a donut?' both CL and RL shift to a LOCATION question. Once the location is established, CL moves on to other attributes while RL keeps asking the same LOCATION question, which leads to failure. Further illustrative examples are given in the supplement.

3.6.3 Analysis of the CL learning process



(a) Lexical diversity

(b) Question diversity

(c) % Games w/ repeated Q's

(d) KL-distance from human

Figure 3.8: Evolution of linguistic factors over 100 training epochs for our GDSE-CL model. Note: lexical and question diversity of the human data fall outside the range in (a) / (b). The same is the case with KL for BL in (d).

In order to better understand the effect of the cooperative learning regime, we trace the evolution of linguistic factors identified above over the CL epochs. As illustrated in Figure 3.8 (a) and (b), through the epochs the CL model learns to use a richer vocabulary and more diverse questions, moving away from the levels achieved by BL and RL, overpassing SL and moving toward humans.

The CL model progressively produces fewer repeated questions within a dialogue, improving over SL in the last few epochs, cf. Figure 3.8 (c). Finally, (d) illustrates the effect of modulo-n training: As the model is trained on generated dialogues, its linguistic output drifts away from the human distribution of question types; every 5^{th} epoch QGen is trained via supervision, which brings the model's behavior closer back to human linguistic style and helps decrease the drift.

3.7 Conclusion

We present a new visually-grounded joint Questioner agent for goal-oriented dialogue. First, we show that our architecture archives 6–9% accuracy improvements over the *GuessWhat?!* baseline system de Vries et al. (2017). This way, we address a foundational limitation of previous approaches that model guessing and questioning separately.

Second, our joint architecture allows us to propose a two-phase cooperative learning approach (CL), which further improves accuracy, results in our overall best model and reaches state-of-the-art results (cf. Section 4.3.2). We compare CL to the system proposed by Strub et al. (2017a) which extends the baseline with reinforcement learning (RL). We find that the two approaches (CL and RL) achieve overall relatively similar task success rates. However, evaluating on task success is only one side of the coin. Finally and most importantly, we propose to pursue an in-depth analysis of the quality of the dialogues by visual conversational agents, which is an aspect often neglected in the literature. We analyze the linguistic output of the two models across three factors (lexical diversity, question diversity, and repetitions) and find them to differ substantially. The CL model uses a richer vocabulary and inventory of questions, and produces fewer repeated questions than RL. In contrast, RL highly relies on asking location questions, which might be explained by a higher reliance on spatial and object-type information was explicitly given to the Guesser and Oracle models. Limiting rewards to task success or other rewards not connected to the language proficiency does not stimulate the model to learn rich linguistic skills, since a reduced vocabulary and simple linguistic structures may be an effective strategy to succeed at the game.

Further, in our joint architecture, we have also incorporated a decision-making component that decides when to stop asking questions which results in less repetitive and more human-like dialogues. This shows the flexibility of the proposed architecture.

Chapter 4

FOIL Diagnostic Dataset and Tasks

The aim of this chapter is to understand whether current language and vision models truly merge the two modalities. To this end, we propose an extension of the MS-COCO dataset, FOIL-COCO, which associates images with both correct and 'foil' captions, that is, descriptions of the image that is highly similar to the original ones, but contain one single mistake ('foil word'). We show that current LaVi models fall into the traps of this data and perform badly on three tasks: a) caption classification (correct vs. foil); b) foil word detection; c) foil word correction. Humans, in contrast, have near-perfect performance on those tasks. We demonstrate that merely utilizing language cues is not enough to model FOIL-COCO and that it challenges the state-of-the-art by requiring a fine-grained understanding of the relation between text and image.[1]

4.1 Introduction

Most human language understanding is grounded in perception. There is thus growing interest in combining information from language and vision in the NLP and AI communities. Multiple models have been proposed to merge

[1] Part of work of this chapter will appear in ACL 2017 as **Ravi Shekhar**, Sandro Pezzelle, Yauhen Klimovich, Aurelie Herbelot, Moin Nabi, Enver Sangineto and Raffaella Bernardi, "**FOIL it! Find One mismatch between Image and Language caption**", *In Proc. of* 55^{th} *Annual Meeting of the Association for Computational Linguistics (ACL), 2017 (Long-Oral).*

language and vision information. Whilst some models have seemed extremely successful on those tasks, it remains unclear how the reported results should be interpreted and what those models are actually learning. It has been shown that co-occurrence based models are performing close to SoA models of different tasks. For example, for VQA task by simple concatenation of language and vision features (Agrawal et al., 2016; Jabri et al., 2016; Zhang et al., 2016; Goyal et al., 2016a) performs very well. In IC task too, Hodosh and Hockenmaier (2016) showed that contrarily to what prior research had suggested, the task is far from been solved, since IC models are not able to distinguish between a correct and incorrect caption.

Such results indicate that in current datasets, *language provides priors* that make LaVi models successful without truly understanding and integrating language and vision. But problems do not stop at biases. Johnson et al. (2017) also point out that current data 'conflate multiple sources of error, making it hard to pinpoint model weaknesses', thus highlighting the need for *diagnostic datasets*. Thirdly, existing IC *evaluation metrics* are sensitive to n-gram overlap and there is a need for measures that better simulate human judgments (Hodosh et al., 2013b; Elliott and Keller, 2014; Anderson et al., 2016).

This work tackles the identified issues by proposing an automatic method for creating a large dataset of real images with *minimal language bias* and some *diagnostic* abilities. Our dataset, FOIL (Find One mismatch between Image and Language caption),[2] consists of images associated with incorrect captions. The captions are produced by introducing one single error (or 'foil') per caption in existing, human-annotated data (Figure 4.1). This process results in a challenging error-detection/correction setting (because the caption is 'nearly' correct). It also provides us with ground truth (we know where the error is) that can be used to measure the performance of current models objectively.

We propose three tasks based on widely accepted evaluation measures: we

[2] The dataset is available from https://foilunitn.github.io/

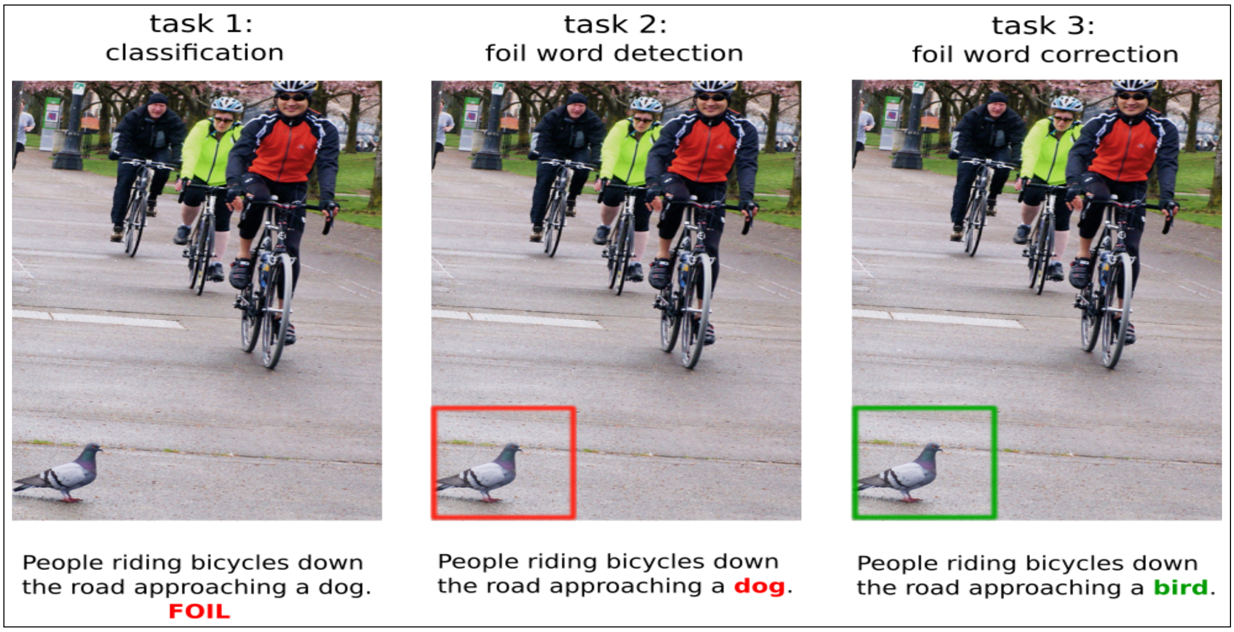

task 1:
classification
task 2:
foil word detection
task 3:
foil word correction
People riding bicycles down the road approaching a dog.
FOIL
People riding bicycles down the road approaching a dog.
People riding bicycles down the road approaching a bird.

Figure 4.1: Is the caption correct or foil (T1)? If it is foil, where is the mistake (T2) and which is the word to correct the foil one (T3)?

test the ability of the system to a) compute whether a caption is compatible with the image (T1); b) when it is incompatible, highlight the mismatch in the caption (T2); c) correct the mistake by replacing the foil word (T3).

The dataset presented in this work (Section 4.2) is built on top of MS-COCO (Lin et al., 2014b), and contains 297,268 datapoints and 97,847 images. We will refer to it as FOIL-COCO. We evaluate two state-of-the-art VQA models: the popular one by Antol et al. (2015), and the attention-based model by Lu et al. (2016), and one popular IC model by Wang et al. (2016). We show that those models perform close to chance level, while humans can perform the tasks accurately (Section 4.3). Section 4.4 provides an analysis of our results, allowing us to diagnose three failures of LaVi models. First, their coarse representations of language and visual input do not encode suitably structured information to spot mismatches between an utterance and the corresponding scene (tested by T1). Second, their language representation is not fine-grained enough to identify the part of an utterance that causes a mismatch with the image as it is (T2). Third, their visual representation is also too poor to spot and name the visual area that corresponds to a captioning error (T3).

4.2 FOIL-COCO Dataset and Task

In this section, we describe how we automatically generate FOIL-COCO datapoints, i.e. image, original and foil caption triples. We used the training and validation set of Microsoft's Common Objects in Context (MS-COCO) dataset (Lin et al., 2014b) (2014 split version) as our starting point. We will first describe the MS-COCO dataset.

MS-COCO Dataset The MS-COCO dataset is created using AMT. First of all, the common categories of the object are decided. These categories are selected such that it is a representative set of all categories, and be relevant to practical applications. Authors first collected object categories from Everingham et al. (2010) and 1200 most frequent word from Sitton (1996). Apart from these, children ages from 4 to 8 were asked to name objects in daily use. This selected list is then pruned using voting based on usefulness for practical applications and their diversity relative to other categories, which resulted in 91 object categories. These object categories (e.g. *dog, elephant, bird, . . .* and *car, bicycle, airplane, . . .*), are from 11 super-categories (*Animal*, *Vehicle*, resp.), with 82 of them having more than 5K labeled instances.

Using these common object categories, the images of the dataset is crawl using an internet search. The internet query is formulated using pairs of objects and images retrieved are performed via scene-based queries (Ordonez et al., 2011). These collected images are then being annotated using a hierarchical labeling approach (Deng et al., 2014). The MS-COCO dataset provides multiple annotations like object bounding box, object segmentation, etc.

For the MS-COCO caption data is collected using AMT. Captions are collected such that it described the important parts of the image and it doesn't just list out the objects in the image. All captions contain at least 8 words. In total there are 123,287 images with captions (82,783 for training and 40,504 for validation).[3] We used the MS-COCO caption to create the FOIL-COCO dataset.

FOIL-COCO Data Generation

Our data generation process consists of four main steps, as described below. The last two steps are illustrated in Figure 4.2.

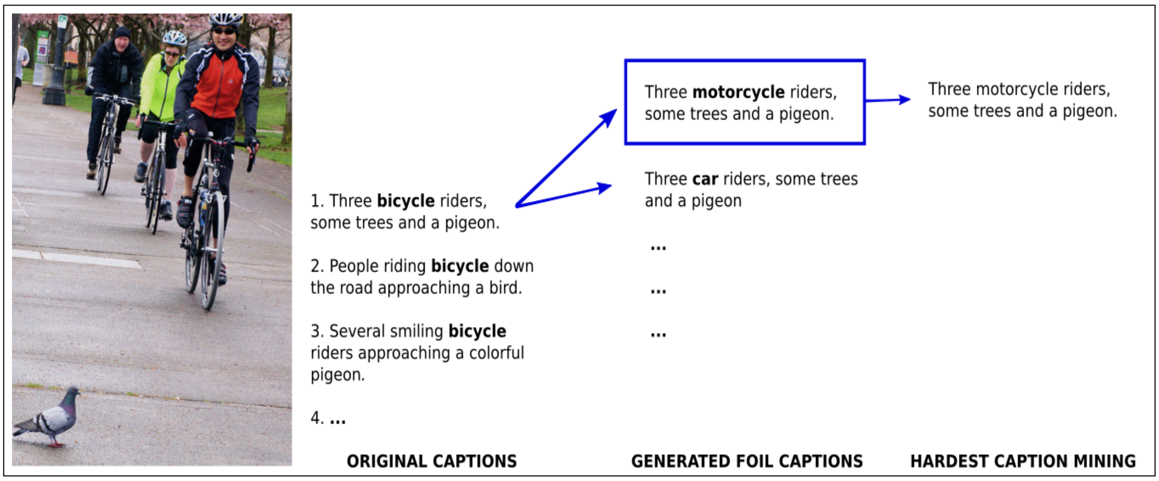

Three **motorcycle** riders, some trees and a pigeon.
Three motorcycle riders, some trees and a pigeon.
Three **car** riders, some trees and a pigeon
1. Three **bicycle** riders, some trees and a pigeon.
2. People riding **bicycle** down the road approaching a bird.
3. Several smiling **bicycle** riders approaching a colorful pigeon.
4. ...
...
...
...
ORIGINAL CAPTIONS
GENERATED FOIL CAPTIONS
HARDEST CAPTION MINING

Figure 4.2: The main aspects of the foil caption generation process. Left column: some of the original COCO captions associated with an image. In bold we highlight one of the target words (bicycle), chosen because it is mentioned by more than one annotator. Middle column: For each original caption and each chosen target word, different foil captions are generated by replacing the target word with all possible candidate foil replacements. Right column: A single caption is selected amongst all foil candidates. We select the 'hardest' caption, according to Neuraltalk model, trained using only the original captions.

1. Generation of replacement word pairs We want to replace one noun in the original caption (the *target*) with an incorrect but similar word (the *foil*). To do this, we take the labels of MS-COCO categories, and we pair together words belonging to the same super-category (e.g., bicycle::motorcycle, bicycle::car,

[3] The MS-COCO test set is not available for download.

bird::dog). We use as our vocabulary 73 out of the 91 MS-COCO categories, leaving out those categories that are multi-word expressions (e.g. traffic light). We thus obtain 472 target::foil pairs.

2. Splitting of replacement pairs into training and testing To avoid the models learning trivial correlations due to replacement frequency, we randomly split, within each supercategory, the candidate target::foil pairs which are used to generate the captions of the training vs. test sets. We obtain 256 pairs, built out of 72 target and 70 foil words, for the training set, and 216 pairs, containing 73 target and 71 foil words, for the test set.

3. Generation of foil captions We would like to generate foil captions by replacing only target words which refer to *visually salient objects*. To this end, given an image, we replace only those target words that occur in more than one MS-COCO caption associated with that image. Moreover, we want to use foils which are *not visually present*, i.e. that refers to visual content not present in the image. Hence, given an image, we only replace a word with foils that are not among the labels (objects) annotated in MS-COCO for that image. We use the images from the MS-COCO training and validation sets to generate our training and test sets, respectively. We obtain 2,229,899 for training and 1,097,012 captions for testing.

4. Mining the hardest foil caption for each image To eliminate possible visual-language dataset bias, out of all foil captions generated in step 3, we select only the hardest one. For this purpose, we need to model the visual-language bias of the dataset. To this end, we use Neuraltalk[4] (Karpathy and Fei-Fei, 2015), one of the state-of-the-art image captioning systems, pre-trained on MS-COCO. Neuraltalk is based on an LSTM which takes as input an image and generates a sentence describing its content. We obtain a neural network $\mathcal{N}$ that implicitly represents the visual-language bias through its weights. We use $\mathcal{N}$ to approximate the conditional probability of a caption C given a dataset

[4] https://github.com/karpathy/neuraltalk

	no. of datapoints	no. unique images	no. of tot. captions	no. target::foil pairs
Train	197,788	65,697	395,576	256
Test	99,480	32,150	198,960	216

Table 4.1: Composition of FOIL-COCO dataset.

T and and an image I ($P(C|I,T)$). This is obtained by simply using the loss $l(C, \mathcal{N}(I))$ i.e., the error obtained by comparing the pseudo-ground truth C with the sentence predicted by $\mathcal{N}$: $P(C|I,T) = 1-l(C, \mathcal{N}(I))$ (we refer to Karpathy and Fei-Fei (2015) for more details on how $l()$ is computed). $P(C|I,T)$ is used to select the hardest foil among all the possible foil captions, i.e. the one with the highest probability according to the dataset bias learned by $\mathcal{N}$. Through this process, we obtain 197,788 and 99,480 original::foil caption pairs for the training and test sets, respectively. None of the target::foil word pairs are filtered out by this mining process.

The final FOIL-COCO dataset consists of 297,268 datapoints (197,788 in training and 99,480 in test set). All the 11 MS-COCO supercategories are represented in our dataset and contain 73 categories from the 91 MS-COCO ones (4.8 categories per supercategory on average). Table 4.1 provides the details of the FOIL-COCO dataset.

FOIL Tasks

Along with the FOIL dataset, we also proposed the following three tasks to test the models, see Figure4.1.

Task 1 (T1): Correct vs. foil classification Given an image and a caption, the model is asked to mark whether the caption is correct or wrong. The aim is to understand whether LaVi models can spot mismatches between their coarse representations of language and visual input.

Task 2 (T2): Foil word detection Given an image and a foil caption, the model has to detect the foil word. The aim is to evaluate the understanding of the system at the word level. In order to systematically check the system's performance with different prior information, we test two different settings: the foil has to be selected amongst (a) only the nouns or (b) all content words in the caption.

Task 3 (T3): Foil word correction Given an image, a foil caption and the foil word, the model has to detect the foil and provide its correction. The aim is to check whether the system's visual representation is fine-grained enough to be able to extract the information necessary to correct the error. For efficiency reasons, we operationalize this task by asking models to select a correction from the set of target words, rather than the whole dataset vocabulary (viz. more than 10K words).

4.3 Experiments and Results

In this section first, we will describe the details of the models used to perform the FOIL tasks. And later, we will provide the results obtained using the FOIL-COCO dataset.

4.3.1 Models Tested

We evaluate both VQA and IC models against our tasks. For the former, we use two of the three models evaluated in Goyal et al. (2016a) against a balanced VQA dataset. For the latter, we use the multimodal bi-directional LSTM, proposed by Wang et al. (2016), and adapted for our tasks.

LSTM + norm I: We use the best performing VQA model in Antol et al. (2015) (deeper LSTM + norm I). To encode the caption, a two stack Long-Short Term Memory (LSTM) is used and caption embedding is obtained by the last hidden

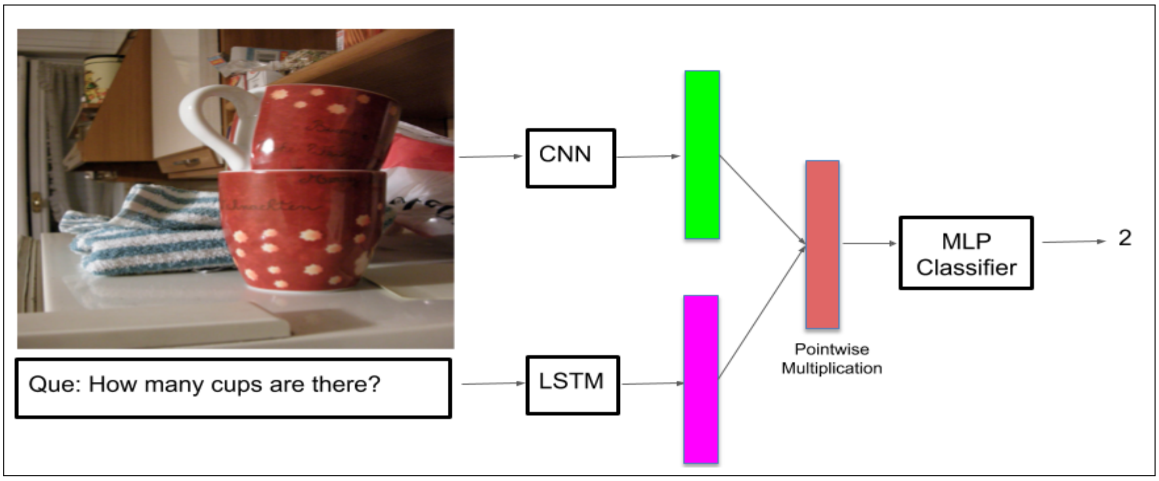


Figure 4.3: VQA Model using LSTM + norm I

layer of the LSTM. To encode the image, VGGNet (Simonyan and Zisserman, 2014) is used. The image embedding is obtained by normalizing the last fully connected layer of VGGNet. Both image embedding and caption embedding are projected into a 1024-dimensional feature space. The combination of these two projected embeddings is performed by a point-wise multiplication. The multi-model representation thus obtained is used for the classification, which is performed by a multi-layer perceptron (MLP) classifier, as shown in Figure 4.3.

HieCoAtt: We use the Hierarchical Co-Attention model proposed by Lu et al. (2016) that co-attends to both the image and the question to solve the task. Authors proposed two types of attention: parallel co-attention and alternating co-attention. In parallel co-attention, image and question is attended simultaneously. In alternating co-attention, attention is sequential alternates between generating some attention over the image and question. We evaluate FOIL task using the 'alternate' version. It does so in a hierarchical way by starting from the word-level, then going to the phrase and then to the entire sentence-level (see Figure 4.4). These levels are combined recursively to produce the distribution over the foil vs. correct captions.

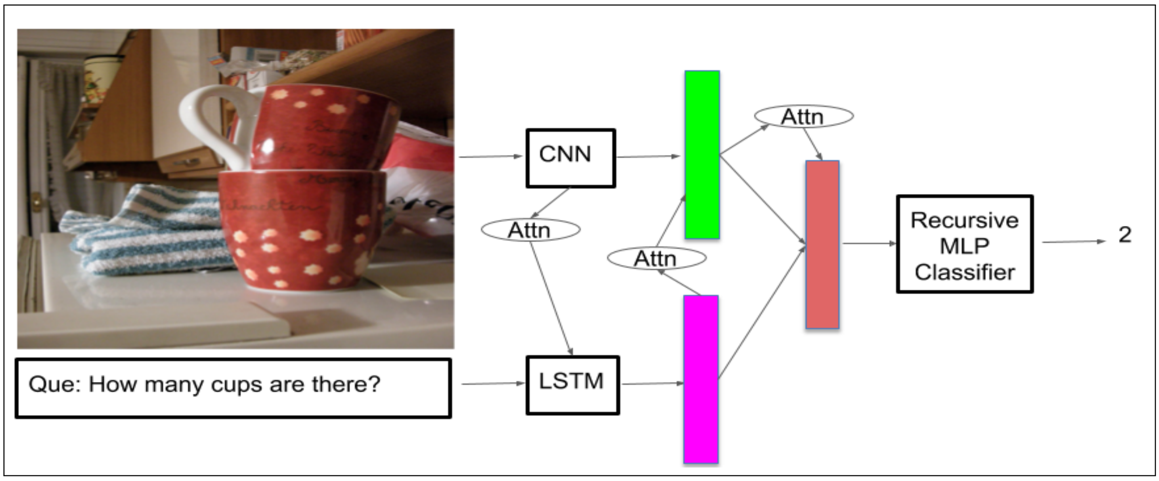


Figure 4.4: VQA Model using Hierarchical Co-Attention model.

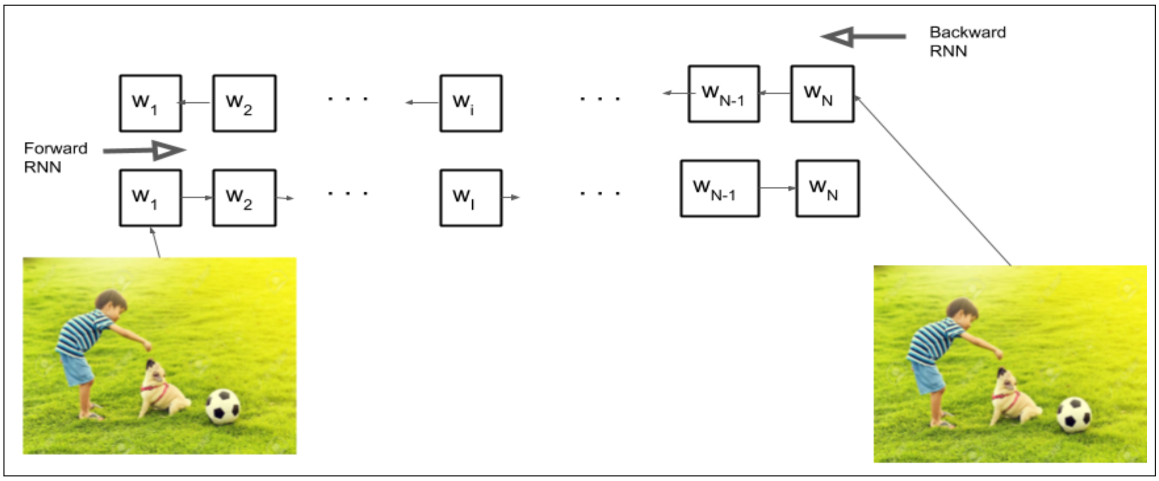


Figure 4.5: IC Model using BiLSTM

IC-Wang: Amongst the IC models, we choose the multimodal bi-directional LSTM (Bi-LSTM) model proposed in Wang et al. (2016). This model predicts a word in a sentence by considering both the past and future context, as sentences are fed to the LSTM in forward and backward order. The model consists of three modules: a CNN for encoding image inputs, a Text-LSTM (T-LSTM) for encoding sentence inputs, a Multimodal LSTM (M-LSTM) for embedding visual and textual vectors to common semantic space and decoding to sentence. The bidirectional LSTM is implemented with two separate LSTM layers.

Baselines: We compare the SoA models above against two baselines. For the classification task, we use a **Blind** LSTM model followed by a fully connected layer and softmax and train it only on captions as input to predict the answer. In addition, we evaluate the **CNN+LSTM** model, where visual and textual features are simply concatenated.

Model's performance on FOIL tasks For the *classification task* (T1), the baselines and VQA models can be applied directly. We adapt the generative IC model to perform the classification task as follows. Given a test image I and a test caption, for each word w_t in the test caption, we remove the word and use the model to generate new captions in which the w_t has been replaced by the word v_t predicted by the model (w_1,...,w_{t-1}, v_t, w_{t-1},...,w_n). We then compare the conditional probability of the test caption with all the captions generated from it by replacing w_t with v_t. When all the conditional probabilities of the generated captions are lower than the one assigned to the test caption the latter is classified as good, otherwise as foil. For the other tasks, the models have been trained on T1. To perform the *foil word detection task* (T2), for the VQA models, we apply the occlusion method. Following Goyal et al. (2016c), we systematically occlude subsets of the language input, forward propagate the masked input through the model, and compute the change in the probability of the answer predicted with the unmasked original input. For the IC model, similarly to T1, we sequentially generate new captions from the foil one by replacing, one by one, the words in it and computing the conditional probability of the foil caption and the one generated from it. The word whose replacement generate the caption with the highest conditional probabilities is taken to be the foil word. Finally, to evaluate the models on the *error correction task* (T3), we apply the linear regression method over all the target words and select the target word which has the highest probability of making that wrong caption correct with respect to the given image.

Upper-bound Using Crowdflower, we collected human answers from 738 native English speakers for 984 image-caption pairs randomly selected from the test set. Subjects were given an image and a caption and had to decide whether it was correct or wrong (T1). If they thought it was wrong, they were required to mark the error in the caption (T2). We collected 2952 judgments (i.e., 3 judgments per pair and 4 judgments per rater) and computed human accuracy in T1 when considering an answer (a) the one provided by at least 2 out of 3 annotators (*majority*) and (b) the one provided by all 3 annotators (*unanimity*). The same procedure was adopted for computing accuracies in T2. Accuracies in both T1 an T2 are reported in Table 4.2 and 4.3 respectively. As can be seen, in the *majority* setting annotators are quasi-perfect in classifying captions (92.89%) and detecting foil words (97.00%). Though lower, accuracies in the *unanimity* setting are still very high, with raters providing the correct answer in 3 out of 4 cases in both tasks. Hence, although we have collected human answers only on rather a small subset of the test set, we believe their results are the representative of how easy the tasks are for humans.

4.3.2 Results

As shown in Tables 4.2, 4.3, 4.4, the FOIL-COCO dataset is challenging. On T1(Table 4.2), for which the chance level is 50.00%, the 'blind', language-only model, does badly with an accuracy of 55.62% (25.04% on foil captions), demonstrating that language bias is minimal. By adding visual information, CNN+LSTM, the overall accuracy increases by 5.45% (7.94% on foil captions.) reaching 61.07% (resp. 32.98%). Both SoA VQA and IC models do significantly worse than humans on both T1 and T2. The VQA systems show a strong bias towards correct captions and poor overall performance. They only identify 34.51% (LSTM +norm I) and 36.38% (HieCoAtt) of the incorrect captions (T1). On the other hand, the IC model tends to be biased toward the foil captions, on which it achieves an accuracy of 45.44%, higher than the VQA models. But the

T1: Classification task			
	Overall	Correct	Foil
Blind	55.62	86.20	25.04
CNN+LSTM	61.07	89.16	32.98
IC-Wang	42.21	38.98	45.44
LSTM + norm I	63.26	**92.02**	34.51
HieCoAtt	**64.14**	91.89	**36.38**
Human (*majority*)	92.89	91.24	94.52
Human (*unanimity*)	76.32	73.73	78.90

Table 4.2: **T1:** Accuracy for the *classification* task, relatively to all image-caption pairs (overall) and by type of caption (correct vs. foil).

overall accuracy (42.21%) is poorer than the one obtained by the two baselines. On the foil word detection task, when considering only nouns as possible foil word, both the IC and the LSTM+norm I models perform close to chance level, and the HieCoAtt performs somewhat better, reaching 38.79%. Similar results are obtained when considering all words in the caption as the possible foil. Finally, the VQA models' accuracy on foil word correction (T3) is extremely low, at 4.7% (LSTM +norm I) and 4.21% (HieCoAtt). The result on T3 makes it clear that the VQA systems are unable to extract from the image representation the information needed to correct the foil: despite being told which element in the caption is wrong, they are not able to zoom into the correct part of the image to provide a correction, or if they are, cannot name the object in that region. The IC model performs better compared to the other models, having an accuracy that is 20,78% higher than chance level.

4.4 Analysis

We performed a mixed-effect logistic regression analysis in order to check whether the behavior of the best performing models in T1, namely the VQA models, can be predicted by various linguistic variables. We included: 1) se-

T2: Foil word detection task		
	nouns	all content words
Chance	23.25	15.87
IC-Wang	27.59	23.32
LSTM + norm I	26.32	24.25
HieCoAtt	**38.79**	**33.69**
Human (*majority*)		97.00
Human (*unanimity*)		73.60

Table 4.3: **T2:** Accuracy for the *foil word detection* task, when the foil is known to be among the nouns only or when it is known to be among all the content words.

T3: Foil word correction task	
	all target words
Chance	1.38
IC-Wang	**22.16**
LSTM + norm I	4.7
HieCoAtt	4.21

Table 4.4: **T3:** Accuracy for the *foil word correction* task when the correct word has to be chosen among any of the target words.

mantic similarity between the original word and the foil (computed as the cosine between the two corresponding word2vec embeddings Mikolov et al. (2013)); 2) frequency of original word in FOIL-COCO captions; 3) frequency of the foil word in FOIL-COCO captions; 4) length of the caption (number of words). The mixed-effect model was performed to get rid of possible effects due to either object supercategory (indoor, food, vehicle, etc.) or target::foil pair (e.g., zebra::giraffe, boat::airplane, etc.). For both LSTM + norm I and HieCoAtt, word2vec similarity, the frequency of the original word, and frequency of the foil word turned out to be highly reliable predictors of the model's response. The higher the values of these variables, the more the models tend to provide the wrong output. That is, when the foil word (e.g. *cat*) is semantically very similar to the original one (e.g. *dog*), the models tend to wrongly classify the caption as 'correct'. The same holds for frequency values. In particular, the higher the frequency of both the original word and the foil one, the more the models fail. This indicates that systems find it difficult to distinguish related concepts at the text-vision interface, and also that they may tend to be biased towards frequently occurring concepts, 'seeing them everywhere' even when they are not present in the image. Caption length turned out to be only a partially reliable predictor in the LSTM + norm I model, whereas it is a reliable predictor in HieCoAtt. In particular, the longer the caption, the harder for the model to spot that there is a foil word that makes the caption wrong.

As revealed by the fairly high variance explained by the random effect related to target::foil pairs in the regression analysis, both models perform very well on some target::foil pairs, but fail on some others (see leftmost part of Table 4.6 for same examples of easy/hard target::foil pairs). Moreover, the variance explained by the random effect related to object supercategory is reported in Table 4.5. As can be seen, for some supercategories accuracies are significantly higher than for others (compare, e.g., 'electronic' and 'outdoor').

In a separate analysis, we also checked whether there was any correlation

Super-category	No. of object	No. of foil captions	Acc. using LSTM + norm I	Acc. using HieCoAtt
outdoor	2	107	2.80	0.93
food	9	10407	22.00	26.59
indoor	6	4911	30.74	27.97
appliance	5	2811	32.72	34.54
sports	10	16276	31.57	31.61
animal	10	21982	39.03	43.18
vehicle	8	16514	34.38	40.09
furniture	5	13625	33.27	33.13
accessory	5	3040	49.53	31.80
electronic	6	5615	45.82	43.47
kitchen	7	4192	38.19	45.34

Table 4.5: Classification Accuracy of foil captions by Super Categories (T1). The No. of the objects and the No. of foil captions refer to the test set. The training set has a similar distribution.

between results and the position of the foil in the sentence, to ensure the models did not profit from any undesirable artifacts of the data. We did not find any such correlation.

To better understand results on T2, we performed an analysis investigating the performance of the VQA models on a different target::foil pairs. As reported in Table 4.6 (right), both models perform nearly perfectly with some pairs and very badly with others. At first glance, it can be noticed that LSTM + norm I is very effective with pairs involving vehicles (*airplane*, *truck*, etc.), whereas HieCoAtt seems more effective with pairs involving animate nouns (i.e. animals), though more in-depth analysis is needed on this point. More interestingly, some pairs that are found to be predicted almost perfectly by LSTM + I norm, namely boat::airplane, zebra::giraffe, and drier::scissors, turn out to be among the Bottom-5 cases in HieCoAtt. This suggests, on the one hand, that the two VQA models use different strategies to perform the task. On the other

Top-5		Bottom-5		Top-5		Bottom-5	
T1: LSTM + norm I				T2: LSTM + norm I			
racket::glove	100	motorcycle::airplane	0	drier::scissors	100	glove::skis	0
racket::kite	97.29	bicycle::airplane	0	zebra::giraffe	88.98	snowboard::racket	0
couch::toilet	97.11	drier::scissors	0	boat::airplane	87.87	donut::apple	0
racket::skis	95.23	bus::airplane	0.35	truck::airplane	85.71	glove::surfboard	0
giraffe::sheep	95.09	zebra::giraffe	0.43	train::airplane	81.93	spoon::bottle	0
T1: HieCoAtt				T2: HieCoAtt			
tie::handbag	100	drier::scissors	0	zebra::elephant	94.92	drier::scissors	0
snowboard::glove	100	fork::glass	0	backpack::handbag	94.44	handbag::tie	0
racket::skis	100	handbag::tie	0	cow::zebra	93.33	broccoli:orange	1.47
racket::glove	100	motorcycle::airplane	0	bird::sheep	93.11	zebra::giraffe	1.96
backpack::handbag	100	train::airplane	0	orange::carrot	92.37	boat::airplane	2.09

Table 4.6: Easiest and hardest target::foil pairs: T1 (caption classification) and T2 (foil word detection).

hand, it shows that our dataset does not contain cases that are a priori easy for any model.

The results of IC-Wang on T3 are much higher than LSTM + norm I and HieCoAtt, although it is outperformed by or is on par with HieCoAtton on T1-T2. Our interpretation is that this behavior is related to the discriminative/generative nature of our tasks. Specifically, T1 and T2 are discriminative tasks and LSTM + norm I and HieCoAtt are discriminative models. Conversely, T3 is a generative task (a word needs to be generated) and IC-Wang is a generative model. It would be interesting to test other IC models on T3 and compare their results against the ones reported here. However, note that IC-Wang is 'tailored' for T3 because it takes as input the whole sentence (minus the word to be generated), while common sequential IC approaches can only generate a word depending on the previous words in the sentence.

As far as human performance is concerned, both T1 and T2 turn out to be extremely easy. In T1, image-caption pairs were correctly judged as correct/wrong in overall 914 out of 984 cases (92.89%) in the *majority* setting. In the *unanimity* setting, the correct response was provided in 751 out of 984 cases (76.32%).

Judging foil captions turns out to be slightly easier than judging correct captions in both settings, probably due to the presence of typos and misspellings that sometimes occur in the original caption (e.g. raters judge as wrong the original caption *People playing ball with a drown and white dog*, where 'brown' was misspelled as 'drown'). To better understand which factors contribute to making the task harder, we qualitatively analyze those cases where all annotators provided a wrong judgment for an image-caption pair. As partly expected, almost all cases where original captions (thus correct for the given image) are judged as being wrong are cases where the original caption is indeed incorrect. For example, a caption using the word 'motorcycle' to refer to a bicycle in the image is judged as wrong. More interesting are those cases where all raters agreed in considering as correct image-caption pairs that are instead foil. Here, it seems that vagueness, as well as certain metaphorical properties of language, are at play: human annotators judged as correct a caption describing *Blue and banana large birds on tree with metal pot* (see Fig 4.6, left), where 'banana' replaced 'orange'. Similarly, all raters judged as correct the caption *A cat laying on a bed next to an opened keyboard* (see Fig 4.6, right), where the cat is instead laying next to an opened laptop.

Focusing on T2, it is interesting to report that among the correctly-classified foil cases, annotators provided the target word in 97% and 73.6% of cases in the *majority* and *unanimity* setting, respectively. This further indicates that finding the foil word in the caption is a rather trivial task for humans.

4.5 Conclusion

We have introduced FOIL-COCO, a large dataset of images associated with both correct and foil captions. The error production is automatically generated, but carefully thought out, making the task of spotting foils particularly challenging. By associating the dataset with a series of tasks, we allow for diagnosing

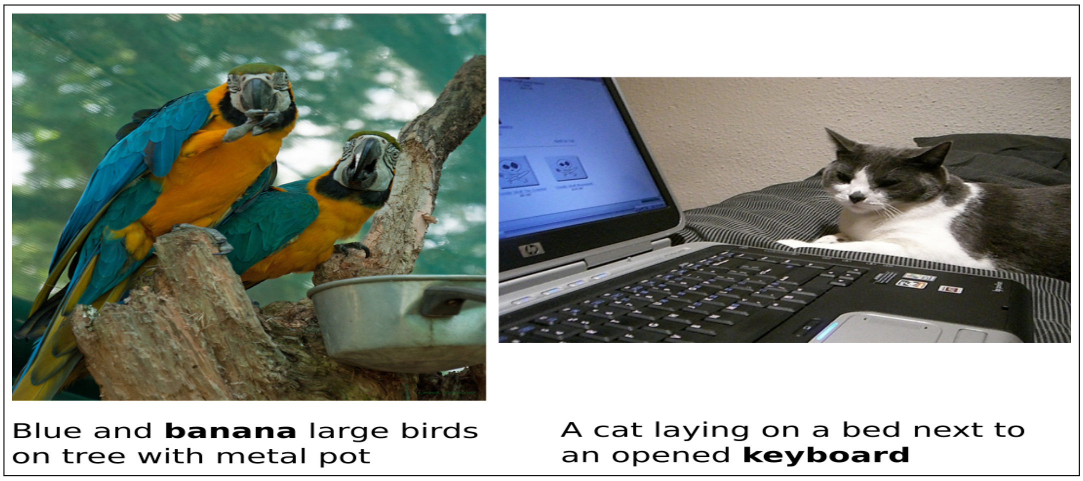

Blue and **banana** large birds on tree with metal pot
A cat laying on a bed next to an opened **keyboard**

Figure 4.6: Two cases of foil image-caption pairs that are judged as correct by all annotators.

various failures of current LaVi systems, from their coarse understanding of the correspondence between text and vision to their grasp of language and image structure.

Our hypothesis is that systems which, like humans, deeply integrate the language and vision modalities, should spot foil captions quite easily. The SoA LaVi models we have tested fall through that test, implying that they fail to integrate the two modalities. To complete the analysis of these results, a further task is needed to detect in the image the area that produces the mismatch with the foil word (the red box around the bird in Figure 4.1.) This extra step would allow us to fully diagnose the failure of the tested systems and confirm what is implicit in our results from task 3: that the algorithms are unable to map particular elements of the text to their visual counterparts.

LaVi models are a great success of recent research, and with the amount of ideas, data and models produced in this stimulating area. With this work, we would like to push the community to think beyond the task success and develop models that can better merge language and vision modalities, instead of merely using one to supplement the other.

Chapter 5

Evaluation of the Encoder of Language and Vision Models

The multimodal models used in the emerging field at the intersection of computational linguistics and computer vision implement the bottom-up processing of the "Hub and Spoke" architecture proposed in cognitive science to represent how the brain processes and combines multi-sensory inputs. In particular, the Hub is implemented as a neural network encoder. We investigate the effect on this encoder of various vision-and-language tasks proposed in the literature: visual question answering, visual reference resolution, and visually grounded dialogue. To measure the quality of the representations learned by the encoder, we use two kinds of analyses. First, we evaluate the encoder pre-trained on the different vision-and-language tasks on an existing *diagnostic task* designed to assess multimodal semantic understanding. Second, we carry out a battery of analyses aimed at studying how the encoder merges and exploits the two modalities.[1]

[1] Part of work of this chapter will appear in IWCS 2019 as
Ravi Shekhar, Ece Takmaz, Raffaella Bernardi, and Raquel Fernández, "**Evaluating the Representational Hub of Language and Vision Models** ", *In Proc. of* 13^{th} *International Conference on Computational Semantics (IWCS), 2019 (Long-Oral).*

5.1 Introduction

In recent years, a lot of progress has been made within the emerging field at the intersection of computational linguistics and computer vision thanks to the use of deep neural networks. The most common strategy to move the field forward has been to propose different multimodal tasks—such as visual question answering (Antol et al., 2015), visual question generation (Mostafazadeh et al., 2016), visual reference resolution (Kazemzadeh et al., 2014), and visual dialogue (Das et al., 2017a)—and to develop task-specific models.

The benchmarks developed so far have put forward complex and distinct neural architectures, but in general they all share a common backbone consisting of an encoder which learns to merge the two types of representation to perform a certain task. This resembles the bottom-up processing in the 'Hub and Spoke' model proposed in Cognitive Science to represent how the brain processes and combines multi-sensory inputs (Patterson and Ralph, 2015). In this model, a 'hub' module merges the input processed by the sensor-specific 'spokes' into a joint representation. We focus our attention on the encoder implementing the 'hub' in artificial multimodal systems, with the goal of assessing its ability to compute multimodal representations that are useful beyond specific tasks.

While current visually grounded models perform remarkably well on the task they have been trained for, it is unclear whether they are able to learn representations that truly merge the two modalities and whether the skill they have acquired is stable enough to be transferred to other tasks. In this paper, we investigate these questions in detail. To do so, we evaluate an encoder trained on different multimodal tasks on an existing *diagnostic task*—FOIL classification task (see Section 4.2)—designed to assess multimodal semantic understanding and carry out in-depth analysis to study how the encoder merges and exploits the two modalities. We also exploit two techniques to investigate the structure of the learned semantic spaces: Representation Similarity Analysis

(RSA) (Kriegeskorte et al., 2008) and Nearest Neighbour overlap (NN). We use RSA to compare the outcome of the various encoders given the same vision-and-language input and NN to compare the multimodal space produced by an encoder with the ones built with the input visual and language embeddings, respectively, which allows us to measure the relative weight an encoder gives to the two modalities.

In particular, we consider three visually grounded tasks: visual question answering (VQA) (Antol et al., 2015), where the encoder is trained to answer a question about an image; visual resolution of referring expressions (ReferIt) (Kazemzadeh et al., 2014), where the model has to pick up the referent object of a description in an image; and GuessWhat (de Vries et al., 2017), where the model has to identify the object in an image that is the target of a goal-oriented question-answer dialogue. We make sure the datasets used in the pre-training phase are as similar as possible in terms of size and image complexity, and use the same model architecture for the three pre-training tasks. This guarantees fair comparisons and the reliability of the results we obtain.

We show that the multimodal encoding skills learned by pre-training the model on GuessWhat and ReferIt are more stable and transferable than the ones learned through VQA. This is reflected in the lower number of epochs and the smaller training data size they need to reach their best performance on the FOIL task. We also observe that the semantic spaces learned by the encoders trained on the ReferIt and GuessWhat tasks are closer to each other than to the semantic space learned by the VQA encoder. Despite these asymmetries among tasks, we find that all encoders give more weight to the visual input than the linguistic one.

5.2 Visually Grounded Tasks and Diagnostic Task

We study three visually grounded tasks: visual question answering (VQA), visual resolution of referring expressions (ReferIt), and goal-oriented dialogue for visual target identification (GuessWhat). While ReferIt was originally formulated as an object detection task (Kazemzadeh et al., 2014), VQA (Antol et al., 2015) and GuessWhat (de Vries et al., 2017) were defined as classification tasks. Here we operationalize the three tasks as retrieval tasks, which makes comparability easier.

- **VQA:** Given an image and a natural language question about it, the model is trained to retrieve the correct natural language answer out of a list of possible answers.
- **ReferIt:** Given an image and a natural language description of an entity in the image, the model is asked to retrieve the bounding box of the corresponding entity out of a list of candidate bounding boxes.
- **GuessWhat:** Given an image and a natural language question-answer dialogue about a target entity in the image, the model is asked to retrieve the bounding box of the target among a list of candidate bounding boxes. The GuessWhat game also involves asking questions before guessing. Here we focus on the guessing task that takes place after the question generation step.

Figure 5.1 (left) exemplifies the similarities and differences among the three tasks. All three tasks require merging and encoding visual and linguistic input. In VQA, the system is trained to make a language-related prediction, while in ReferIt it is trained to make visual predictions. GuessWhat includes elements of both VQA and ReferIt, as well as specific properties: The system is trained to make a visual prediction (as in ReferIt) and it is exposed to questions (as in VQA); but in this case the linguistic input is a coherent sequence of visually grounded questions and answers that follow a goal-oriented strategy and that have been produced in an interactive setting.

To evaluate the multimodal representations learned by the encoders of the models trained on each of the three tasks above, we leverage the FOIL task

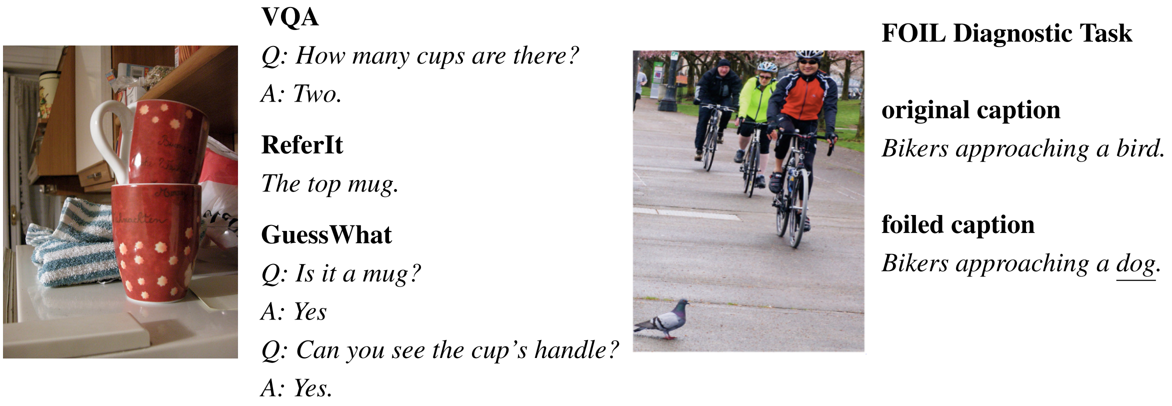


Figure 5.1: Illustrations of the three visually-grounded tasks (left) and the diagnostic task (right).

(concretely, task 1 introduced in Section 4.2), a binary classification task designed to detect semantic incongruence in visually grounded language.

- **FOIL (diagnostic task):** Given an image and a natural language caption describing it, the model is asked to decide whether the caption faithfully describes the image or not, i.e., whether it contains a foiled word that is incompatible with the image. Figure 5.1 (right) shows an example.

5.3 Model Architecture and Training

In cognitive science, the hub module of Patterson and Ralph (2015) receives representations processed by sensory-specific spokes and computes a multimodal representation out of them. All our models have a common core that resembles this architecture, while incorporating some task-specific components. This allows us to investigate the impact of specific tasks on the multimodal representations computed by the representational hub, which is implemented as an encoder. Figure 5.2 shows a diagram of the shared model components, which we explain in detail below.

5.3.1 Shared components

To facilitate the comparison of the representations learned via the different tasks we consider, we use pre-trained visual and linguistic features to process the input given to the encoders. This provides a common initial base across models and diminishes the effects of using different datasets for each specific task (the datasets are described in Section 5.4).

Visual and language embeddings For the visual input, we use ResNet152 features He et al. (2016), which yield state of the art performance in image classification tasks and can be computed efficiently. For the linguistic input, we use Universal Sentence Encoder (USE) vectors Cer et al. (2018) since they yield near state-of-the-art results on several NLP tasks and are suitable both for short texts (such as the descriptions in ReferIt) and longer ones (such as the dialogues in GuessWhat[2]).

Encoder As shown in Figure 5.2, ResNet152 visual features ($V \in \mathbb{R}^{2048\times 1}$) and USE linguistic features ($L \in \mathbb{R}^{512\times 1}$) are input in the model and passed through fully connected layers that project them onto spaces of the same dimensionality. The projected representations (V_p and L_p) are concatenated, passed through a linear layer, and then through a *tanh* activation function, which produces the final encoder representation h:

$$h = \tanh\left(W \cdot [V_p;\ L_p]\right) \tag{5.1}$$

where $W \in \mathbb{R}^{1024\times 1024}$, $V_p \in \mathbb{R}^{512\times 1}$, $L_p \in \mathbb{R}^{512\times 1}$, and $[\cdot;\cdot]$ represents concatenation.

[2] The dialogues in the GuessWhat?! dataset consists of 4.93 question-answer pairs on average de Vries et al. (2017).

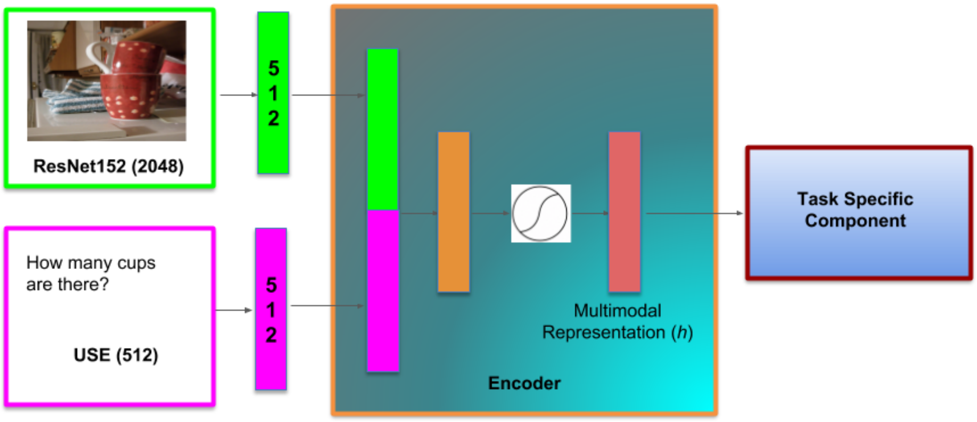


Figure 5.2: General model architecture. The encoder receives as input visual (ResNet152) and linguistic (USE) embeddings and merges them into a multimodal representation (h). This is passed on to a task-specific component: an MLP in the case of the pre-training retrieval tasks and a fully connected layer in the case of the FOIL classification task.

5.3.2 Task-specific components

The architecture described above is shared by all the models we experiment with, which thus differ only with respect to their task-specific component.

Pre-training task component For the three tasks we consider, the final encoder representation h is given to a Multi-Layer Perceptron (MLP), which generates either a language embedding (VQA model) or a visual embedding (ReferIt and GuessWhat model). The three task-specific models are trained with a cosine similarity loss, which aims to get the generated embedding closer to the ground truth embedding and farther away from any other embeddings.

FOIL task component To evaluate the encoder representations learned by the pre-trained models, the task-specific MLPs are replaced by a fully connected layer, which is trained on the FOIL task using a cross-entropy loss. We train the FOIL task component using the following settings:

- **Random**$_2$ The encoder weights are randomly initialized and the FOIL classifier layer is untrained. This provides a lower-bound baseline with random performance.

- **Random** The encoder weights are randomly initialized and then frozen while the FOIL classifier layer is trained on the FOIL task. This provides a strong baseline that is directly comparable to the task-specific setting explained next.
- **Pre-trained (VQA, ReferIt, GuessWhat)** The encoder weights are initialized with the Random setting's seeds and the model is trained on each of the tasks. The weights of the task-specific encoders are then frozen and the FOIL classifier layer is trained on the FOIL task. With this setting, we are able to diagnose the transfer and encoding properties of the pre-trained tasks.
- **Fully trained on FOIL** The encoder weights are initialized with the Random setting's seeds. Then the full model is trained on the FOIL task, updating the weights of the projected vision and language layers, the encoder, and the FOIL layer. This provides the upper bound on the FOIL classification performance, as the entire model is optimized for this task from the start.

5.4 Experimental Setup

We provide details on the data sets and the implementation settings we use in our experiments.

For the three visually grounded tasks, we use the VQA.v1 dataset by Antol et al. (2015), the RefCOCO dataset by Yu et al. (2016a), and the GuessWhat?! dataset by de Vries et al. (2017) as our starting point. All these datasets have been developed with images from MS-COCO Lin et al. (2014a).

QA pairs of VQA.v1 dataset(Antol et al. (2015) is collect using human annotators. Annotators are instructed to ask interesting, diverse and requires image to answer. For every question, two types of answers are collected: open-ended and multiple choice. The open-ended answer is selected via majority annotators agreement. While for the multiple choice answers, 18 candidate answers are created for each question. Candidate answers are created based on correctness, plausibility, and popularity. The correct answers are selected by majority agreement. The plausible answers are collected by asking the question without showing the image. And the popular answers are those which are the most popular answers in the dataset. For every image on average 3 QA pairs are collected.

The referring expression in the RefCOCO dataset (Yu et al., 2016a) is collected using AMT in an interactive setting. The first player is asked to write the referring expression for a given image and bounding box of a target object. The target object is selected such that it has multiple instances of that object in the image. After writing the referring expressions, the second player is provided the image and corresponding referring expression to select the object. If the selected object is matching with the target object that referring expression is selected for that image and target object. If both players do their job correctly, they are rewarded and their role is swapped. The dataset is collected such that it also consists of appearance-based description, not only location based. The GuessWhat?! dataset is also collect using AMT, in which to guess a target object in the image multiple round QA-pairs is used, see Section 3.2 for details.

Common Dataset Since all the datasets are collected independently and has a different set of images and word distribution. Our goal is to create a common dataset to minimizing the difference in the dataset. In this direction, we first construct common image datasets for by taking the intersection of the images in the three original datasets. This results in a total of 14,458 images. An image can be part of several data points, i.e, it can be paired with more than one linguistic input. Indeed, the 14,458 common images correspond to 43,374 questions for the VQA task, 104,227 descriptions for the ReferIt task, and 35,467 dialogues for the GuessWhat task.

To obtain datasets of equal size per task that are as similar as possible, we filter the resulting data points according to the following procedure:

1. For each image, we check how many linguistic items are present in the three datasets and fix the minimum number (k) to be our target number of linguistic items paired with that image.
2. We select n data points where the descriptions in ReferIt and dialogues in GuessWhat concern the same target object (with $n \leq k$).
3. Among the n data points selected in the previous step, we select the m data points in VQA where the question or the answer mention the same target object (computed by string matching).
4. We make sure all the images in each task-specific dataset are paired with exactly k linguistic items; if not, we select additional ones randomly until this holds.

This results in a total of 30,316 data points per dataset: 14,458 images shared across datasets, paired with 30,313 linguistic items. We randomly divided this *common image* dataset into training and validation sets at the image level. The training set consists of 13,058 images (paired with 27,374 linguistic items) and the validation set of 1,400 images (paired with 2,942 linguistic items). Table 5.1 provides an overview of the datasets.

As mentioned in Section 5.2, we operationalize the three tasks as retrieval tasks where the goal is to retrieve the correct item out of a set of candidates. In the VQA.v1 dataset (multiple choice version), there are 18 candidate answers per question. In GuessWhat?! there are on average 18.71 candidate objects per dialogue, all of them appearing in the image. We take the same list of candidate objects per image for the ReferIt task.

	common image datasets		**FOIL dataset**		
	training	validation	training	validation	testing
# images	13,058	1,400	63,240	13,485	20,105
# language	27,374	2,942	358,182	37,394	126,232

Table 5.1: Statistics of the datasets used for the pre-training tasks and the FOIL task.

FOIL dataset The FOIL dataset consists of image-caption pairs from MS-COCO and pairs where the caption has been modified by replacing a noun in the original caption with a foiled noun, such that the foiled caption is incongruent with the image—see Figure 5.1 for an example and Section 4.2 for further details on the construction of the dataset.[3] The dataset contains 521,808 captions (358,182 in training, 37,394 in validation and 126,232 in test set) and 96,830 images (63,240, 13,485 and 20,105, in training, validation and test set, respectively) – see Table 5.1. All the images in the test set do not occur either in the FOIL training and validation set, nor in the common image dataset described above and used to pre-train the models.

Implementation details All models are trained using supervised learning with ground truth data. We use the same parameters for all models: batch size of 256 and Adam optimizer Kingma and Ba (2014b) with learning rate 0.0001. All the parameters are tuned on the validation set. Early stopping is used while training, i.e., training is stopped when there is no improvement on the validation loss for 10 consecutive epochs or a maximum of 100 epochs, and the best model is taken based on the validation loss.

[3] Madhysastha et al. (2018) found that an earlier version of the FOIL dataset was biased. We have used the latest version of the dataset available at `https://foilunitn.github.io/`, which does not have this problem.

5.5 Results and Analysis

We carry out two main blocks of analyses: one exploiting FOIL as diagnostic task and the other one investigating the structure of the semantic spaces produced by the pre-trained encoders when receiving the same multimodal inputs.

Before diving into the results of these analyses, we evaluate the three task-specific models on the tasks they have been trained for. Since these are retrieval tasks we compute Mean Rank, obtaining 2.84 (VQA), 3.32 (ReferIt) and 4.14 (GuessWhat) on the validation sets. The models learn to perform the task reasonably well, as shown by the fact that Precision@1 results for each model are above chance: 0.14 for VQA (chance 0.055), 0.12 for ReferIt, and 0.08 for GuessWhat (chance 0.05 for the latter two).

5.5.1 Analysis via diagnostic task

In this first analysis, we assess the quality of the multimodal representations learned by the three multimodal tasks considered in terms of their potential to perform the FOIL task, i.e., to spot semantic (in)congruence between an image and a caption. Besides comparing the models with respect to task accuracy, we also investigate how they learn to adapt to the FOIL task over training epochs, how much data they need to reach their best performance, and how confident they are about the decisions they make.

FOIL accuracy Table 5.2 shows accuracy results on the FOIL task for the different training settings described in Section 5.3.2. We report accuracy for the task overall, as well as accuracy on detecting original and foiled captions. As expected, the Random$_2$ setting yields chance performance (≈50% overall, with a surprisingly strong preference for classifying captions as foiled). The model fully trained on FOIL achieves an accuracy of 67.59%. This confirms that the

FOIL task is challenging, as shown in Chapter 4, even for models that are optimized to solve it. The Random setting, where a randomly initialized encoder is trained on the FOIL task, yields 53.79% accuracy overall – higher than the chance lower bound by Random$_2$, but well below the upper bound set by the fully trained model.

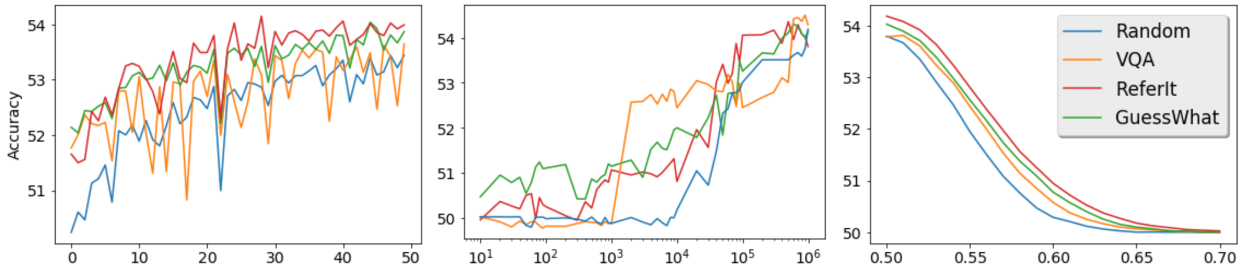
The key results of interest for our purposes in this paper are those achieved by the models where the encoder has been pre-trained on each of the three multimodal tasks we study. We observe that, like the Random encoder, the pre-trained encoders achieve results well below the upper bound. The VQA encoder yields result comparable to Random, while ReferIt and GuessWhat achieve slightly higher results: 54.02% and 54.18%, respectively. This trend is much more noticeable when we zoom into the accuracy results on original vs. foiled captions. All models (except Random$_2$) achieve lower accuracy on the foil class than on the original class. However, the GuessWhat encoder performs substantially better than the rest: Its foil accuracy is not only well above the Random encoder, but also around 2% points over the fully trained model (49.34% vs. 47.52%). The ReferIt encoder also performs reasonably well (on a par with the fully trained model), while the VQA encoder is closer to Random.

This suggests that the ReferIt and the GuessWhat encoders do learn a small degree of multimodal understanding skills that can transfer to new tasks. The VQA encoder, in contrast, seems to lack this ability by and large.

Learning over time In order to better understand the effect of the representations learned by the pre-trained encoders, we trace the evolution of the FOIL classification accuracy over time, i.e., over the first 50 training epochs. As shown in Figure 5.3a, all the pre-trained models start with higher accuracy than the Random model. This shows that the encoder is able to transfer knowledge from the pre-trained tasks to some extent. The Random model takes around 10 epochs to catch up and after that, it does not manage to improve much. The evolution

	overall	original	foiled
Random$_2$	49.99	0.282	99.71
Random	53.79	65.33	42.25
VQA	53.78	66.09	41.48
ReferIt	54.02	60.39	47.66
GuessWhat	54.18	59.02	49.34
Fully FOIL	67.59	87.66	47.52

Table 5.2: Accuracy on the FOIL task for the best model of each training setting.



(a) Training epochs.

(b) Size of FOIL training set (log scaled).

(c) AUC indicating confidence.

Figure 5.3: Comparisons among the pre-trained encoders and the randomly initialized encoder, regarding their accuracy over training epochs, with varying data size, and across different decision thresholds.

of the accuracy achieved by the ReferIt and GuessWhat encoders is relatively smooth, i.e., it increases progressively with further training epochs. The one by the VQA model, in contrast, is far less stable.

Size of FOIL training data Next, we evaluate how the accuracy achieved by the models changes when varying the size of the FOIL training set. By controlling the amount of training data, we can better tease apart whether the performance of the pre-trained models is due to the quality of the encoder representations or simply to the amount of training the models undergo on the FOIL task itself. Figure 5.3b gives an overview. The GuessWhat encoder has a clear advantage when very little training data is available, while the other encoders start at chance level. Both GuessWhat and ReferIt increase their accuracy relatively smoothly as more data is provided, while for the VQA model there is a big jump in accuracy once enough FOIL data is available. Again, this suggests that the representations learned by the GuessWhat encoder are of somewhat higher quality, with more transferable potential.

Confidence Finally, we analyse the confidence of the models by measuring their Area Under the Curve (AUC). We gradually increase the classification threshold from 0.5 to 0.7 by an interval of 0.01. This measures the confidence of the classifier in making a prediction. As shown in Figure 5.3c, all models have rather low confidence (when the threshold is 0.7 they are all at chance level). The Random model exhibits the lowest confidence, while the ReferIt model is slightly more confident in its decisions than the rest, followed by the GuessWhat model.

5.5.2 Analysis of the multimodal semantic spaces learned by the encoders

In this section, we analyse the encoders by comparing the similarity of the multimodal spaces they learn and by comparing the learned multimodal spaces to the visual and linguistic representations they receive as input in terms on nearest neighbours.

Representation similarity analysis Representation Similarity Analysis (RSA) is a technique from neuroscience Kriegeskorte et al. (2008) that has been recently leveraged in computational linguistics, for example to compare the semantic spaces learned by artificial communicating agents Bouchacourt and Baroni (2018). It compares different semantic spaces by comparing their internal similarity relations, given a common set N of input data points. Each input $k \in N$ is processed by an encoder for a given task Ti, producing vector h^k_{Ti}.

Let H^N_{Ti} be the set of vector representations created by the encoder of Ti for all the items in N; and let H^N_{Tj} be the corresponding set of representations by the encoder of task Tj. These two semantic spaces, H^N_{Ti} and H^N_{Tj}, are not directly comparable as they have been produced independently. RSA remedies this by instead comparing their structure in terms of internal similarity relations. By computing cosine similarity between all pairs of vectors within each semantic space, we obtain a vector of cosine similarities per space, which captures its internal structure. These similarity vectors have identical dimensionality, namely $N(N-1)/2$ values, and hence can be directly compared by computing Spearman correlation between them. The resulting RSA scores (corresponding to the aforementioned Spearman correlation coefficients) tell us the extent to which the two sets of representations are structurally similar.

The outputs of the encoders are compared when the same set of inputs is given. We give as input 5,000 data points from the FOIL test set, randomly sampled from only the ones with original captions and containing unique images, and compare the representations produced by the encoders under investigation. Figure 5.4 shows that the semantic space produced by the encoder fully trained on FOIL is rather different from all the other models, and that the VQA semantic space is very similar to the one produced by the randomly initialized encoder.

Nearest neighbour overlap We analyze the encoder representations using nearest neighbor overlap. Collell and Moens (2018) proposed this measure to compare the structure of functions that map concepts from an input to a target space. It is defined as the number of k nearest neighbors that two paired vectors share in their respective semantic space. For instance, if $k = 3$ and the 3 nearest neighbours of the vector for 'cat' v_{cat} in space V are $\{v_{dog}, v_{tiger}, v_{lion}\}$, and those of the vector of 'cat' z_{cat} in space Z are $\{v_{mouse}, v_{tiger}, v_{lion}\}$, the nearest neighbour overlap (NN) is 2. The value is then normalized with respect to the

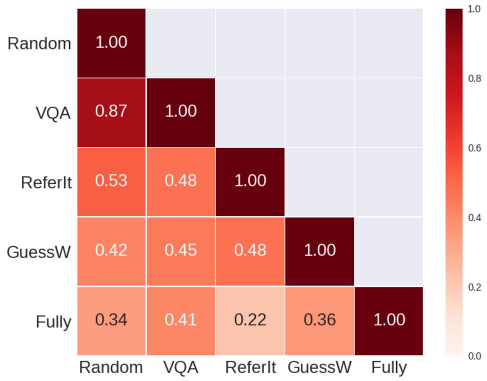


Figure 5.4: RSA scores indicating degree of structural similarity between the multimodal semantic spaces produced by the various encoders when receiving 5,000 data points from the FOIL test set consisting of unique images paired with their original captions.

	$k = 1$		$k = 10$	
	ResNet152	USE	ResNet152	USE
Random	0.829	0.363	0.876	0.365
VQA	0.638	0.350	0.703	0.386
ReferIt	0.7541	0.346	0.780	0.366
GuessWhat	0.658	0.329	0.689	0.359
Fully FOIL	0.171	0.254	0.246	0.291

Table 5.3: Average nearest neighbour overlap between the encoder multimodal representations and the ResNet152 and USE embeddings, respectively.

number of data points and the number of k nearest neighbors.

We take the encoder to be a mapping function from each of the modality-specific representations to the multimodal space, and we use the NN measure to investigate whether the structure of the multimodal space produced by the encoder is closer to the visual ResNet152 embeddings or to the linguistic USE embeddings given as input. We use simple visual and language inputs, namely, objects and the word corresponding to their object category. We considered the 80 object categories of MS-COCO (e.g., dog, car, etc.) and obtain their USE representations. We built their visual ResNet152 embedding by selecting 100 images for each category from MS-COCO, and then compute their average. We computed the NN by setting $k = 1$ and $k = 10$. The results, given in Table 5.3, show that the multimodal spaces learned by all the models (except the model with the encoder fully trained on the FOIL task) are much closer to the visual input space than to the linguistic one.

5.6 Conclusion

Our goal in this chapter has been to evaluate the quality of the multimodal representations learned by an encoder—the core module of all the multimodal models used currently within the language and vision community—which resembles the cognitive representational hub described by Patterson and Ralph (2015). Furthermore, we investigated the transfer potential of the encoded skills, taking into account the amount of time (learning epochs) and training data the models need to adapt to a new task and with how much confidence they make their decisions. We studied three multimodal tasks, where the encoder is trained to answer a question about an image (VQA), pick up the object in an image referred to by a description (ReferIt), and identify the object in an image that is the target of a goal-oriented question-answer dialogue (GuessWhat). To carry out this analysis, we have evaluated how the pre-trained models perform on the FOIL diagnostic task, designed to check the model's ability to detect semantic incongruence in visually grounded language.

Our analysis shows that the VQA task is easier to learn (the model achieves a rather high Mean Rank precision). However, the multimodal encoding skills it learns are less stable and transferable than the ones learned through the ReferIt and GuessWhat tasks. This can be seen by a large amount of data the model has to be exposed to in order to learn the FOIL classification task and by the unstable results over training epochs. None of the models transfers their encoding skills with high confidence, but again the VQA model does it to a lower extent.

The RSA analysis confirms the higher similarity of the multimodal spaces generated by the ReferIt and GuessWhat encoders and the high similarity between the VQA space and the space produced by the randomly initialized encoder. From the NN analysis, it appears that for all models (except for the one fully trained on the FOIL task) the visual modality has a higher weight than the linguistic one in the construction of the multimodal representations.

These results could be due to subtle parallelisms with the diagnostic task: ReferIt and GuessWhat may resemble some aspects of FOIL, since these three tasks revolve around objects (the foiled word is always a noun), while arguably the VQA task is more diverse as it contains questions about, e.g., actions, attributes, or scene configurations. In future work, it would be interesting to evaluate the models on different diagnostic datasets that prioritize skills other than object identification.

Chapter 6

Conclusion

We have witnessed rapid advancement in the fields related to AI, like Computer Vision, Natural Language Processing, Machine Learning etc, thanks to deep neural networks. Using DNNs, performance on multiple tasks are getting close to the human performance. For language and vision tasks, image captioning performance is very close to human evaluation. Similarly, for VQA models are getting closer to human performance. In this thesis, we took a step back and analysis the performance of these models.

Concretely, in Chapter 3 we develop a joint model for GuessWhat?!, a task-oriented visual dialogue, to address one of the fundamental problems in the guessing game. We have also incorporated a decision-making module in the pipeline to stop the dialogue when it has enough information to perform the task. To improve the performance, we have incorporated the co-operative learning paradigm into the architecture. Our proposed model trained with the co-operative learning setting performed close to models trained with reinforcement learning. To further analyze the different model's outputs, we proposed a thorough analysis based on the different linguistic features of the dialogue. We show that even though the accuracy of the models is similar, the languages learned by those models are very different.

In Chapter 4, we introduce a diagnostic dataset (FOIL) to show that state-of-the-art models are far from merging complimentary language and vision information. The FOIL dataset is created using MS-COCO image captioning dataset by replacing one correct word with the foil word. We proposed three FOIL tasks to evaluate the models. First, given an image and a caption the model has to classify the latter to be right or wrong (classification task); this task tests the overall representation learning of the model. Second, given an image and a foil caption, the model has to spot the foil word in it (detection task); this task tests the fine-grained understanding at language level using the image. The third task, given an image, the foil caption and the foil word in it, the model has to propose the correct word (correct task); this task aims to evaluate the fine-grained understanding of both language and vision. All these tasks test the different level of information merging of the model.

Finally, in Chapter 5, we exploit the FOIL dataset to evaluate the representation learned by a model trained on three language and vision tasks: VQA, ReferIt and GuessWhat. To overcome the difference in the dataset, we created a common dataset. For all tasks, we trained the model in a retrieval setting to have a better comparison among models. We show that on the FOIL classification task, all the model performance is close to each other, while the learning strategy is different. Further, using representation similarity analysis, we show that models representation is far from each other. Also, based on the nearest neighbor analysis, we find that multi-model representation learned by the model is structurally similar to the visual representation.

Coming back to Figure 1.1, by looking at the image and corresponding description, we could not only answer questions like *Was Mary's office close?*. We could also infer things like *Only authorized person is allowed in the office.* or *The door requires a manual key, not electronic key.* These type of inference from the image and given description can be only drawn when the model has all the fine-grained information about the image and text and it is able to form the correspondence between those. Along with that, it requires knowledge of the world like *authorized person* is related to *having key of the door*. Current state-of-the-art models are far from reaching this level of maturity.

Bibliography

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *International Conference on Learning Representations (ICLR)*.

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 4971–4980.

Afra Alishahi, Marie Barking, and Grzegorz Chrupała. 2017. Encoding of phonology in a recurrent neural model of grounded speech. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics, pages 368–378. https://doi.org/10.18653/v1/K17-1037.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *In Proceedings of the European Conference on Computer Vision (ECCV)*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *International Conference on Computer Vision (ICCV)*.

Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M Blei, and Michael I Jordan. 2003. Matching words and pictures. *Journal of machine learning research* 3(Feb):1107–1135.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal oriented dialog. In *Proceedings of ICLR*.

Diane Bouchacourt and Marco Baroni. 2018. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 981–985. http://aclweb.org/anthology/D18-1119.

Okko Buß, Timo Baumann, and David Schlangen. 2010. Collaborating on utterances with a spoken dialogue system using an isu-based approach to incremental dialogue management. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, pages 233–236.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* .

Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. 2017. Evaluating visual conversational agents via cooperative human-ai games. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.

Xinlei Chen and C Lawrence Zitnick. 2015. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 2422–2431.

Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. 2017. Attend to you: Personalized image captioning with context sequence memory networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 895–903.

Herbert H Clark. 1996. *Using Language*. Cambridge University Press.

Guillem Collell and Marie-Francine Moens. 2018. Do neural network crossmodal mappings really bridge modalities? *arXiv preprint arXiv:1805.07616* .

Alexis Conneau, German Kruszewski, Guillaume Lampl, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of ACL*.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017a. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. 2017b. Learning cooperative visual dialog agents with deep reinforcement learning. In *International Conference on Computer Vision (ICCV)*.

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. 2017. Guesswhat?! Visual object discovery through multi-modal dialogue. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. 2014. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pages 3099–3102.

Nan Ding, Sebastian Goodman, Fei Sha, and Radu Soricut. 2016. Understanding image and text simultaneously: a dual vision-language machine comprehension task. *arXiv preprint arXiv:1612.07833* .

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 2625–2634.

Pinar Duygulu, Kobus Barnard, Joao FG de Freitas, and David A Forsyth. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European conference on computer vision*. Springer, pages 97–112.

Desmond Elliott and Frank Keller. 2013. Image description using visual dependency representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pages 1292–1302.

Desmond Elliott and Frank Keller. 2014. Comparing automatic evaluation measures for image description. In *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: Short Papers*. pages 452–457.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and

Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88(2):303–338.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 1473–1482.

Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision*. Springer, pages 15–29.

Zhe Gan, Yu Cheng, Ahmed EI Kholy, Linjie Li, Jingjing Liu, and Jianfeng Gao. 2019. Multi-step reasoning via recurrent dual attention for visual dialog. *arXiv preprint arXiv:1902.00579* .

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in Neural Information Processing Systems*. pages 2296–2304.

Fabrizio Ghigi, Maxine Eskenazi, M Ines Torres, and Sungjin Lee. 2014. Incremental dialog processing in a task-oriented dialog. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. pages 1440–1448.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. pages 2672–2680.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016a. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *arXiv preprint arXiv:1612.00837* .

Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. 2016b. Towards transparent ai systems: Interpreting visual question answering models. *arXiv preprint arXiv:1608.08974* .

Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. 2016c. Towards Transparent AI Systems: Interpreting Visual Question Answering Models . In *In Proceedings of ICML Visualization Workshop*.

Abhinav Gupta and Larry S Davis. 2008. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *European conference on computer vision*. Springer, pages 16–29.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Micah Hodosh and Julia Hockenmaier. 2013. Sentence-based image description with scalable, explicit models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pages 294–300.

Micah Hodosh and Julia Hockenmaier. 2016. Focused evaluation for image description with binary forced-choice tasks. In *Proceedings of the 5th Workshop on Vision and Language (VL'16)*.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013a. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47:853–899.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013b. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47:853–899.

Hexiang Hu, Ishan Misra, and Laurens van der Maaten. 2019. Binary image selection (bison): Interpretable evaluation of visual grounding. *arXiv preprint arXiv:1901.06595* .

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 4555–4564.

A. Jabri, A. Joulin, and L.J.P. van der Maaten. 2016. Revisiting visual question answering baselines. In *Proceedings of the European Conference on Computer Vision (ECCV)*. pages 727–739.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of CVPR 2017*.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2015. Lingusitic analysis of multi-modal recurrent neural networks. In *Proceedings of the Fourth Workshop on Vision and Language*. pages 8–9.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics* 43(4):761–780.

Kushal Kafle and Christopher Kanan. 2016. Answer-type prediction for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 4976–4984.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3128–3137.

Andrej Karpathy, Armand Joulin, and Fei Fei F Li. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*. pages 1889–1897.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. 2014. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*.

Casey Kennington and David Schlangen. 2016. Supporting spoken assistant systems with a graphical user interface that signals incremental understanding and prediction state. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. pages 242–251.

Hatim Khouzaimi, Romain Laroche, and Fabrice Lefevre. 2015. Optimising turn-taking strategies with reinforcement learning. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. pages 315–324.

Dongho Kim, Catherine Breslin, Pirros Tsiakoulis, M Gašić, Matthew Henderson, and Steve Young. 2014. Inverse reinforcement learning for micro-turn management. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Diederik P Kingma and Jimmy Ba. 2014a. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Diederik P Kingma and Jimmy Ba. 2014b. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*. pages 3276–3284.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. 2008. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience* 2:4.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123(1):32–73.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. pages 1097–1105.

Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. 2013. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(12):2891–2903.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Sang-Woo Lee, Yu-Jung Heo, and Byoung-Tak Zhang. 2018. Answerer in questioner's mind: Information theoretic approach to goal-oriented visual dialog. In *Advances in Neural Information Processing Systems*. pages 2584–2594.

Sang-Woo Lee, Yujung Heo, and Byoung-Tak Zhang. 2017. Answerer in questioner's mind for goal-oriented visual dialogue. In *NIPS Workshop on Visually-Grounded Interaction and Language (ViGIL)*.

David Lewis. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic* 8(1):339–359.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 110–119. http://www.aclweb.org/anthology/N16-1014.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of EMNLP*.

Siming Li, Girish Kulkarni, Tamara L Berg, Alexander C Berg, and Yejin Choi. 2011. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 220–228.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, Dollar, P., and C. L. Zitnick. 2014a. Microsoft COCO: Common objects in context. In *Proceedings of ECCV (European Conference on Computer Vision)*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014b. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*. Springer, pages 740–755.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Trans-*

actions of the Association for Computational Linguistics 4:521–535. http://aclweb.org/anthology/Q16-1037.

Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. 2017. Referring expression generation and comprehension via attributes. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 4856–4864.

Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, and Dhruv Batra. 2017. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *Neural Inforamtion Processing Systems (NIPS) 2017*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *In Proceedings of NIPS 2016*. `https://github.com/jiasenlu/HieCoAttenVQA`.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2018. Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 7219–7228.

Pranava Madhysastha, Josiah Wang, and Lucia Specia. 2018. Defoiling foiled image captions. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. New Orleans, LA.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*. pages 1682–1690.

Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. 2015. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 1–9.

Ramesh Manuvinakurike, David DeVault, and Kallirroi Georgila. 2017. Using reinforcement learning to model incrementality in a fast-paced dialogue game. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. pages 331–341.

Ramesh Manuvinakurike, Casey Kennington, David DeVault, and David Schlangen. 2016. Real-time understanding of complex discriminative scene descriptions. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. pages 232–241.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 11–20.

Daniela Massiceti, N Siddharth, Puneet K Dokania, and Philip HS Torr. 2018. Flipdial: A generative model for two-way visual dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 6097–6105.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios P. Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. *CoRR* abs/1701.08251. http://arxiv.org/abs/1701.08251.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1802–1813. http://www.aclweb.org/anthology/P16-1170.

Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. 2016. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*. Springer, pages 792–807.

Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *Advances in neural information processing systems*. pages 1143–1151.

Maike Paetzel, Ramesh R. Manuvinakurike, and David DeVault. 2015. "so, which one is it?" the effect of alternative incremental architectures in a high-performance game-playing agent. In *SIGDIAL Conference*. pages 77–86.

Dong Huk Park, Daylen Yang, Akira Fukui, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Conference on Empirical Methods in Natural Language Processing*.

Karalyn Patterson and Matthew A. Lambon Ralph. 2015. *Neurobiology of Language*, Elsevier, chapter The Hub-and-Spoke Hypothesis of Semantic Memory.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*. pages 2641–2649.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics, pages 139–147.

Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. Multi-task learning for joint language understanding and dialogue state tracking. In *Proceedings of SIGdial*.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015a. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*. pages 2935–2943.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015b. Image question answering: A visual semantic embedding model and a new dataset. *Proc. Advances in Neural Inf. Process. Syst* 1(2):5.

Hannes Rieser and David Schlangen. 2011. Introduction to the special issue on incremental processing in dialogue. *Dialogue & Discourse* 1:1–10.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 618–626.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Ravi Shekhar, Tim Baumgärtner, Aashish Venkatesh, Elia Bruni, Raffaella Bernardi, and Raquel Fernandez. 2018. Ask no more: Deciding when to guess in referential visual dialogue. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018) (to appear)*. ArXiv:1805.06960.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*. Springer, pages 746–760.

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

Rebecca Sitton. 1996. *Rebecca Sitton's Spelling Sourcebook*. R. Sitton.

Richard Socher and Li Fei-Fei. 2010. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pages 966–973.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015a. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, pages 553–562.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015b. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of NAACL-HLT*. pages 196–205.

Robert Stalnaker. 1978. Assertion. In P. Cole, editor, *Pragmatics*, New York Academic Press, volume 9 of *Syntax and Semantics*.

Florian Strub, Harm de Vries, Jeremie Mary, Bilal Piot, Aaron Courville, and

Olivier Pietquin. 2017a. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *Joint Conference on Artificial Intelligence*.

Florian Strub, Harm De Vries, Jeremie Mary, Bilal Piot, Aaron Courville, and Olivier Pietquin. 2017b. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *International Joint Conference on Artificial Intelligence*.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Vancouver, Canada, pages 217–223. http://aclweb.org/anthology/P17-2034.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*. pages 3104–3112.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2015. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817* .

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2015. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361* .

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3156–3164.

Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. 2016. Image captioning with deep bidirectional LSTMs. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, pages 988–997.

Jason Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of ACL-2017*. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*. pages 404–413.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*. pages 2397–2406.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044* 2(3):5.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2015. Stacked attention networks for image question answering. *arXiv preprint arXiv:1511.02274* .

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. *arXiv preprint arXiv:1603.03925* .

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2:67–78.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE* 101(5).

Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. 2015. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2461–2469.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016a. Modeling context in referring expressions. In *European Conference on Computer Vision*. Springer, pages 69–85.

Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2016b. Training an adaptive dialogue policy for interactive learning of visually grounded word meanings. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. page 339.

Junjie Zhang, Qi Wu, Chunhua Shen, Jian Zhang, Jianfeng Lu, and Anton van den Hengel. 2018. Goal-oriented visual question generation via intermediate rewards. In *Proceedings of the European Conference of Computer Vision (ECCV)*.

Kelly W. Zhang and Samnuel R. Bowman. 2018. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, pages 359–361.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 5014–5022.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state traching and management using deep reinforcement learning. In *Proceedings of SIGDIAL-2016*.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 2921–2929.

Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2015. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167* .

Yan Zhu, Shaoting Zhang, and Dimitris Metaxas. 2017. Interactive reinforcement learning for object grounding via self-talking. In *NIPS Workshop on Visually-Grounded Interaction and Language (ViGIL)*.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 4995–5004.

C Lawrence Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3009–3016.

C Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. 2013. Learning the visual interpretation of sentences. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 1681–1688.

Appendix A

Joint Learning

A.1 Analysis

We provide further details and examples related to the analyses carried out in Section 7.2 of the main paper.

A.1.1 Question classification

The analysis by question type is based on a classification on a set of keywords. These keywords have been annotated using information in the MS-COCO dataset plus manual annotation. First, we created the possible question categories by inspecting the human dialogues. As explained in the paper, the resulting categories are ENTITY, subdivided into SUPER-CATEGORY and OBJECT, and ATTRIBUTE, sub-divided into COLOR, LOCATION, SHAPE, SIZE, TEXTURE and ACTION. We exploited the super-category and object annotations from MS-COCO. To further enrich these annotations, we manually annotated the words in the human dialogues that occur at least 40 times in the training and testing sets. In Table A.2, we report the complete list of keywords highlighting those obtained from COCO. Algorithm 1 provides the pseudo-code of the question classification heuristics we used. Table A.1 provides some examples of the resulting classification.

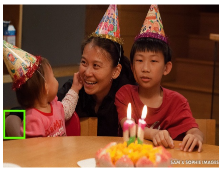
Algorithm 1: Question Classification.

```
Input  : Question and annotated words (from Table 1).
Output: Question Classification
Let $Q = \{w_1...w_t...w_n\}$ denotes all the words for the given Question ;
Let $Color$, $Shape$, $Size$ , $Texture$, $Location$, $Action$, $Object$, $Super$ denotes all
  words present in the 'Color', 'Shape', 'Size', 'Texture', 'Location', 'Action', 'Object'
  and 'Super-category' respectively ;
Let $Q_{cat}$ a Empty List ;
for $\forall w_k \in Q$ do
    if $w_k \in Color$ then
        $Q_{cat} \leftarrow Q_{cat} + color$ ;          // Append color to $Q_{cat}$
        break
    end
end
for $\forall w_k \in Q$ do
    if $w_k \in Shape$ then
        $Q_{cat} \leftarrow Q_{cat} + shape$ ;          // Append shape to $Q_{cat}$
        break
    end
end
for $\forall w_k \in Q$ do
    if $w_k \in Size$ then
        $Q_{cat} \leftarrow Q_{cat} + size$ ;           // Append size to $Q_{cat}$
        break
    end
end
for $\forall w_k \in Q$ do
    if $w_k \in Texture$ then
        $Q_{cat} \leftarrow Q_{cat} + texture$ ;        // Append texture to $Q_{cat}$
        break
    end
end
for $\forall w_k \in Q$ do
    if $w_k \in Location$ then
        $Q_{cat} \leftarrow Q_{cat} + location$ ;       // Append location to $Q_{cat}$
        break
    end
end
```

```
for $\forall w_k \in Q$ do
    if $w_k \in Action$ then
        $Q_{cat} \leftarrow Q_{cat} + action$ ;                // Append action to $Q_{cat}$
        break
    end
end
if $Q_{cat}$ is EMPTY then
    for $\forall w_k \in Q$ do
        if $w_k \in Object$ then
            $Q_{cat} \leftarrow object$ ;                      // Assign object to $Q_{cat}$
            break
        end
    end
end
if $Q_{cat}$ is EMPTY then
    for $\forall w_k \in Q$ do
        if $w_k \in Super$ then
            $Q_{cat} \leftarrow super$ ;              // Assign super-category to $Q_{cat}$
            break
        end
    end
end
if $Q_{cat}$ is EMPTY then
    $Q_{cat} \leftarrow not - classified$ ;          // Assign not-classified to $Q_{cat}$
end
return $Q_{cat}$
```

Question	Question type
is it a *basket*?	OBJECT
is it a *human*?	SUPER-CATEGORY
is it the person in the *middle*?	LOCATION
is the person wearing a *white* shirt?	COLOR
is it the *round* table?	SHAPE
is it the *little* plate?	SIZE
is he wearing a *striped* shirt?	TEXTURE

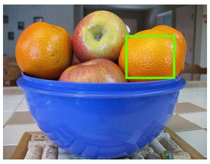
Table A.1: Examples of questions from the human dialogues with keywords in italics and the types assigned through our classification procedure.



Humans	[*success*]	**GDSE-CL**	[*success*]	**RL**	[*failure*]
1. is it a person?	no	1. is it a person?	no	1. is it a person?	no
2. is it a birthday hat?	no	2. is it a cake?	no	2. is it a food?	no
3. is it the candles?	no	3. is it a cup?	no	3. is it a table?	no
4. is it a donut?	no	4. is it a table?	no	4. is it in left?	yes
5. is it the cake?	no	5. is it a chair?	yes	5. is it in left?	yes
6. is it a piece of clothing?	no	6. is it the one on the left side of the table?	yes	6. is it in front?	yes
7. is it a chair?	yes	7. is it the one which is sitting?	yes	7. is it in top?	no
8. is it the one the baby is sitting in?	yes	8. is it just behind the girl?	yes	8. is it in top?	yes
9. Is it the red part?	yes				

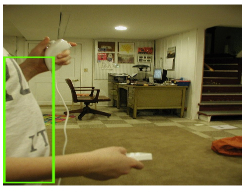
A.1.2 Quality of the Dialogues

A.1, A.2, A.3 and A.4 provide further examples of the games played by the models.



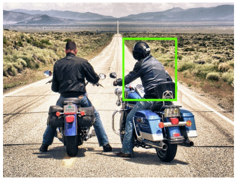
Humans [*success*]		**GDSE-CL** [*success*]		**RL** [*failure*]	
1. is it a fruit?	yo	1. is it an orange?	yes	1. is it food?	yes
2. is it the apple?	no	2. is it in the front?	yes	2. is it one of the oranges?	yes
3. is it the left orange?	no	3. is it an apple?	no	3. is it in left?	no
4. is it the smaller orange on the right?	no	4. is it all the orange?	yes	4. is it in middle?	no
5. is it the bigger orange on the right?	yes	5. can you see most of it?	yes	5. is it in top?	yes
		6. is it on the left side of the picture?	no	6. is it in top?	yes
		7. is it the top right corner?	yes	7. is it in top?	yes
		8. so it is an apple touching the apple?	yes	8. is it in top?	yes

Figure A.1: Game examples in which GDSE-CL succeeds and RL fails at guessing the target object (in the green box). In both examples, GDSE-CL asks further attribute-specific questions. In the top example, *chair* is not that visible so the RL model does not ask a question about it, after Q3 it just asks location questions. In contrast, GDSE-CL after getting *chair*(Q5) it tries to ask questions to distinguish amongst chairs. Similarly, in the bottom example, after getting *orange* (Q2), RL keeps asking simplistic location questions. The GDSE-CL model, after getting *orange* (Q1), tries to ask attribute questions along with location questions.



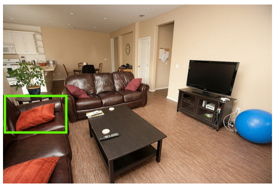
Humans [*success*]		**GDSE-CL** [*success*]		**RL** [*success*]	
1. is it in the background?	no	1. is it a person?	yes	1. is it a person?	yes
2. is it a person?	yes	2. is it the man in the white shirt?	yes	2. is it in foreground?	yes
		3. is it the whole man?	yes	3. is it in left?	yes
		4. does he have a yellow shirt?	yes	4. is it in right?	no
		5. is he on the right side?	no	5. is it in middle?	yes
		6. is he on the left side?	yes	6. is it in right?	no
		7. his face is not visible?	yes	7. is it in middle?	yes
		8. is his face visible?	yes	8. is it in right?	no

Figure A.2: Game example in which both GDSE-CL and RL succeed at guessing the target object (green box), using a different strategy. Here, we have only one instance of the target object super-category *person*, and both models are able to guess the object correctly. While RL keeps asking location questions (Q3-8), GDSE-CL asks questions about color, location, visibility. Even after getting wrong Oracle answer to a question (Q8), it is able to guess the correct object based on previous evidence.



Humans	[*success*]	**GDSE-CL**	[*failure*]	**RL**	[*success*]
1. is it a person?	yes	1. is it a person?	yes	1. is it a person?	yes
2. are they wearing a helmet?	yes	2. is it the man on the motorcycle?	yes	2. is it in foreground?	yes
		3. is it the whole person?	yes	3. is it in left?	no
		4. is it a part of the man?	no	4. is it in right?	yes
		5. is the whole body shown?	yes	5. is it in right?	yes
		6. can i see his whole body?	yes	6. is it in right?	yes
		7. is it the man on top of the bike?	yes	7. is it in right?	yes
		8. is it the whole person?	yes	8. is it in right?	yes

Figure A.3: Game example in which GDSE-CL fails and RL succeeds at guessing the target object (green box). Here, there are only two instances of the target object super-category *person*. The RL strategy of asking location-based questions works well here.



Humans	[*success*]	**GDSE-CL**	[*failure*]	**RL**	[*failure*]
1. is it a sofa?	no	1. is it furniture?	yes	1. is it a person?	no
2. it is green in color?	no	2. is it a couch?	no	2. is it a computer?	no
3. it is a clock?	no	3. a chair then?	yes	3. is it a couch?	no
4. is it a tv?	no	4. the blue one?	yes	4. is it in left?	yes
5. it is blue color?	no	5. the one on the bottom?	no	5. is it in front?	no
6. it is a pillow?	no	6. the left one?	yes	6. in top	yes
7. it is a chair?	yes	7. the whole thing?	yes	7. is it in top?	yes
8. it is black color?	yes	8. just the backrest?	yes	8. is it in top?	yes

Figure A.4: Game examples in which both GDSE-CL and RL fail. The presence of multiple instances of similar target objects (category *chair*) in a similar context makes the task very difficult to the models.

	ENTITY
SUPER-CATEGORY	*‘person’, ‘vehicle’, ‘outdoor’, ‘animal’, ‘accessory’, ‘sports’, ‘kitchen’, ‘food’, ‘furniture’, ‘electronic’, ‘appliance’, ‘indoor’, ‘utensil’*, ‘human’, ‘cloth’, ‘cloths’, ‘clothing’, ‘people’. ‘persons’
OBJECT	*‘bicycle’, ‘car’, ‘motorcycle’, ‘airplane’, ‘bus’, ‘train’, ‘truck’, ‘boat’, ‘traffic light’, ‘fire hydrant’, ‘stop sign’, ‘parking meter’, ‘bench’, ‘bird’, ‘cat’, ‘dog’, ‘horse’, ‘sheep’, ‘cow’, ‘elephant’, ‘bear’, ‘zebra’, ‘giraffe’, ‘backpack’, ‘umbrella’, ‘handbag’, ‘tie’, ‘suitcase’, ‘frisbee’, ‘skis’, ‘snowboard’, ‘sports ball’, ‘kite’, ‘baseball bat’, ‘baseball glove’, ‘skateboard’, ‘surfboard’, ‘tennis racket’, ‘bottle’, ‘wine glass’, ‘cup’, ‘fork’, ‘knife’, ‘spoon’, ‘bowl’, ‘banana’, ‘apple’, ‘sandwich’, ‘orange’, ‘broccoli’, ‘carrot’, ‘hot dog’, ‘pizza’, ‘donut’, ‘cake’, ‘chair’, ‘couch’, ‘potted plant’, ‘bed’, ‘dining table’, ‘toilet’, ‘tv’, ‘laptop’, ‘mouse’, ‘remote’, ‘keyboard’, ‘cell phone’, ‘microwave’, ‘oven’, ‘toaster’, ‘sink’, ‘refrigerator’, ‘book’, ‘clock’, ‘vase’, ‘scissors’, ‘teddy bear’, ‘hair drier’, ‘toothbrush’*, ‘meter’, ‘bear’, ‘cell’, ‘phone’, ‘wine’, ‘glass’, ‘racket’, ‘baseball’, glove’, ‘hydrant’, ‘drier’, ‘kite’, sofa’, ‘fork’, ‘adult’, ‘arms’, ‘baby’, ‘bag’, ‘ball’, ‘bananas’, ‘basket’, ‘bat’, ‘batter’, ‘bike’, ‘birds’, ‘board’, ‘body’, ‘books’, ‘bottles’, ‘box’, ‘boy’, ‘bread’, ‘brush’, ‘building’, ‘bunch’, ‘cabinet’, ‘camera’, ‘candle’, ‘cap’, ‘carrots’, ‘cars’, ‘cart’, ‘case’, ‘catcher’, ‘cell phone’, ‘chairs’, ‘child’, ‘chocolate’, ‘coat’, ‘coffee’, ‘computer’, ‘controller’, ‘counter’, ‘cows’, ‘cupboard’, ‘cups’, ‘curtain’, ‘cycle’, ‘desk’, ‘device’, ‘dining table’, ‘dish’, ‘doll’, ‘door’, ‘dress’, ‘driver’, ‘equipment’, ‘eyes’, ‘fan’, ‘feet’, ‘female’, ‘fence’, ‘fire’, ‘flag’, ‘flower’, ‘flowers’, ‘foot’, ‘frame’, ‘fridge’, ‘fruit’, ‘girl’, ‘girls’, ‘glasses’, ‘guy’, ‘guys’, ‘hair drier’, ‘handle’, ‘hands’, ‘hat’, ‘helmet’, ‘house’, ‘jacket’, ‘jar’, ‘jeans’, ‘kid’, ‘kids’, ‘lady’, ‘lamp’, ‘leg’, ‘legs’, ‘luggage’, ‘machine’, ‘male’, ‘man’, ‘meat’, ‘men’, ‘mirror’, ‘mobile’, ‘monitor’, ‘mouth’, ‘mug’, ‘napkin’, ‘pan’, ‘pants’, ‘paper’, ‘pen’, ‘picture’, ‘pillow’, ‘plant’, ‘plate’, ‘player’, ‘players’, ‘pole’, ‘pot’, ‘purse’, ‘rack’, ‘racket’, ‘road’, ‘roof’, ‘screen’, ‘shelf’, ‘shelves’, ‘shirt’, ‘shoe’, ‘shoes’, ‘short’, ‘shorts’, ’shoulder’, ’signal’, ’sign’, ’silverware’, ’skate’, ’ski’, ’sky’, ’snow’, ’soap’, ’speaker’, ’stairs’, ’statue’, ’stick’, stool’, ‘stove’, ‘street’, ‘suit’, ‘sunglasses’, ‘suv’, ‘teddy’, ‘tennis’, ‘tent’, ‘tomato’, ‘towel’, ‘tower’, ‘toy’, ‘traffic’, ‘tray’, ‘tree’, ‘trees’, ‘t-shirt’, ‘tshirt’, ‘vegetable’, ‘vest’, ‘wall’, ‘watch’, ‘wheel’, ‘wheels’, ‘window’, ‘windows’, ‘woman’, ‘women’

Table A.2: Lists of keywords used to classify questions with the corresponding class according to Algorithm 1. Words in *italics* come from COCO object category/super-category.

ATTRIBUTES	
COLOR	‘white’, ‘red’, ‘black’, ‘blue’, ‘green’, ‘yellow’, ‘orange’, ‘brown’, ‘pink’, ‘grey’, ‘gray’, ‘dark’, ‘purple’, ‘color’, ‘colored’, ‘colour’, ‘blond’, ‘beige’, ‘bright’
SIZE	‘small’, ‘little’, ‘long’, ‘large’, ‘largest’, ‘big’, ‘tall’, ‘smaller’, ‘bigger’, ‘biggest’, ‘tallest’
TEXTURE	‘metal’, ‘silver’, ‘wood’, ‘wooden’, ‘plastic’, ‘striped’, ‘liquid’
SHAPE	‘circle’, ‘rectangle’, ‘round’, ‘shape’, ‘square’, ‘triangle’
LOCATION	‘1st’, ‘2nd’, ‘third’, ‘3’, ‘3rd’, ‘four’, ‘4th’, ‘fourth’, ‘5’, ‘5th’, ‘five’, ‘first’, ‘second’, ‘last’, ‘above’ , ‘ across’ , ‘after’, ‘around’ , ‘at’ , ‘away’ , ‘back ’ , ‘ background’ , ‘before’ , ‘behind’ , ‘below’ , ‘beside’ , ‘between’ , ‘bottom ’ , ‘ center’ , ‘close’ , ‘closer’ , ‘closest’ , ‘corner’ , ‘directly’ , ‘down’ , ‘edge’ , ‘end’ , ‘entire’ , ‘facing’ , ‘far’ , ‘farthest’ , ‘floor’ , ‘foreground’ , ‘from’ , ‘front’ , ‘furthest’ , ‘ground’ , ‘hidden’ , ‘in’ , ‘inside ’ , ‘ left ’ , ‘ leftmost ’ , ‘ middle ’ , ‘ near ’ , ‘ nearest ’ , ‘ next’ , ‘next to’ , ‘off’ , ‘on’ , ‘out’ , ‘outside ’ , ‘ over ’ , ‘ part ’ , ‘ right ’ , ‘ rightmost’ , ‘row’ , ‘side’ , ‘smaller’ , ‘top’ , ‘towards’ , ‘under’ , ‘ up’ , ‘ upper’ , ‘ with’

Table A.3: Lists of keywords used to classify questions with the corresponding class according to Algorithm 1. Words in *italics* come from COCO object category/super-category.