

University of Trento
CIMEC Doctoral School in Cognitive and Brain Sciences
Track Language, Interaction and Computation

Transfer Learning and Attention Mechanisms in a Multimodal Setting

Claudio Greco

Supervisor:

Raffaella Bernardi

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Cognitive and Brain Sciences

Abstract

Humans are able to develop a solid knowledge of the world around them: they can leverage information coming from different sources (e.g., language, vision), focus on the most relevant information from the input they receive in a given life situation, and exploit what they have learned before without forgetting it. In the field of Artificial Intelligence and Computational Linguistics, replicating these human abilities in artificial models is a major challenge. Recently, models based on pre-training and on attention mechanisms, namely pre-trained multimodal Transformers, have been developed. They seem to perform tasks surprisingly well compared to other computational models in multiple contexts. They simulate a human-like cognition in that they supposedly rely on previously acquired knowledge (transfer learning) and focus on the most important information (attention mechanisms) of the input. Nevertheless, we still do not know whether these models can deal with multimodal tasks that require merging different types of information simultaneously to be solved, as humans would do. This thesis attempts to fill this crucial gap in our knowledge of multimodal models by investigating the ability of pre-trained Transformers to encode multimodal information; and the ability of attention-based models to remember how to deal with previously-solved tasks. With regards to pre-trained Transformers, we focused on their ability to rely on pre-training and on attention while dealing with tasks requiring to merge information coming from language and vision. More precisely, we investigate if pre-trained multimodal Transformers are able to understand the internal structure of a dialogue (e.g., organization of the turns); to effectively solve complex spatial questions requiring to process different spatial elements (e.g., regions of the image, proximity between elements, etc.); and to make predictions based on complementary multimodal cues (e.g., guessing the most plausible action by leveraging the content of a sentence and of an image). The results of this thesis indicate that pre-trained Transformers outperform other models. Indeed, they are able to some extent to integrate complementary multimodal information; they manage to pinpoint both the relevant turns in a dialogue and the most important regions in an image. These results suggest that pre-training and attention play a key role in pre-trained Transformers' encoding. Nevertheless, their way of processing information cannot be considered as human-like. Indeed, when compared to humans, they struggle (as non-pre-trained models do) to understand negative answers, to merge spatial information in difficult questions, and to predict actions based on complementary linguistic and visual cues. With regards to attention-based models, we found out that these kinds of models tend to forget what they have learned in previously-solved tasks. However, training these models on easy tasks before more complex ones seems to mitigate this catastrophic forgetting phenomenon. These results indicate that, at least in this context, attention-based models (and, supposedly, pre-trained Transformers too) are sensitive to tasks' order. A better control of this variable may therefore help multimodal models learn sequentially and continuously as humans do.

Acknowledgements

I would like to express my deepest gratitude to my thesis supervisor, Raffaella Bernardi. She gave me all the support I needed throughout this whole winding journey, despite all the responsibilities she constantly deals with. Her interdisciplinary knowledge, academic expertise, great honesty and devotion to Research have been valuable models for me. I learned and am still learning from her about the implications of being a researcher.

A heartfelt thank you goes to Raquel Fernández and Marco Baroni, my Oversight Committee, and to Gemma Boleda and Stephen Clark, the reviewers of this manuscript. All their precious comments and suggestions have been crucial not only to improve the quality of my work, but also to develop my academic reflection.

Thanks also to the great and efficient Leah Mercanti. Without her suggestions and reminders, I would not have been able to keep up with all the administrative requirements and deadlines which are an important part of a PhD program.

To the “Gruppo della Mensa” and especially to Ludovica, Simone, Flavio, Stefano, Alessandro, Madalina, Sandro: you probably do not know how important you have been in these 4 years. You made me laugh and you helped me to “reorder” my thinkings when I felt down. I feel so lucky to have met you all.

Thanks to my long-time friends Giulio, Francesco, Federica and Marco. You have made my trips to Bari legendary! Thank you Fabrizio for being my best friend and my rock since always. I do not know how I would be without your constant support.

And thanks and endless thanks, Alberto. Nothing of this work would have been possible without your friendship, your professional and personal support and your endless patience and positive mood. Thank you for the co-working, the coffee breaks (without coffee), the endless voice messages and all the great time spent together. Just thank you for being always there!

I would like to say thank you also to my aunts and uncles. Their constant support has been essential. And thank you nonne and nonni. You are always with me.

And, above all, THANK YOU mamma, papà, Simone, Dima and Dexter. I owe you everything (good) I am and I have achieved.

For you, Vera, no words would be enough to thank you for all the affection and support you give to me and to my work every day of my life. You mean everything to me. This work simply would have not been possible without you and it belongs to you too.

Contents

Abstract	i
1 Introduction	3
2 Background	6
2.1 Artificial Neural Networks	6
2.1.1 Natural Language Processing	8
2.1.2 Computer Vision	14
2.2 Transfer Learning	18
2.2.1 Natural Language Processing	19
2.2.2 Computer Vision	21
2.3 Multimodal Learning	22
2.3.1 Fusion by product or concatenation	23
2.3.2 Fusion by cross-modal attention	24
2.4 Models	25
2.4.1 RoBERTa	26
2.4.2 LXMERT	26
2.5 Tasks and Datasets	27
2.5.1 CLEVR	28
2.5.2 GuessWhat?!	29

3	Pre-Trained Transformers Encoding the History of a Visual Dialogue	34
3.1	Introduction	34
3.2	Related Work	36
3.3	Dataset	38
3.4	Models	38
3.4.1	Language-only Encoders	38
3.4.2	Multimodal Encoders	39
3.5	Experiments	40
3.5.1	Task Success	40
3.5.2	Are Models Sensitive to the Strategy Seen during Training?	41
3.5.3	The Role of the Last Question	42
3.5.4	How Attention is Distributed across Turns	43
3.5.5	Qualitative Evaluation	43
3.6	Conclusion	45
3.7	Summary	46
4	Pre-trained Transformers Grounding Different Spatial Questions	47
4.1	Introduction	48
4.2	Related Work	51
4.3	Models	52
4.4	The Dataset	53
4.5	Experiments	56
4.5.1	Evaluation by Question Type	56
4.5.2	Evaluation on Spatial Questions	57
4.6	Qualitative Analysis	57
4.6.1	Error Analysis	58
4.6.2	LXMERT’s Attention	60
4.7	Conclusion	62
4.8	Summary	62

5	Pre-Trained Transformers Encoding Positive and Negative Answers	63
5.1	Introduction	64
5.2	Related Work	67
5.3	Task and Dataset	69
5.4	Models	71
5.4.1	Language-only Encoders	72
5.4.2	Multimodal Encoders	73
5.5	Experiments on the Full Test set	74
5.5.1	Accuracy results	74
5.5.2	Guesser’s Probability Distribution	76
5.5.3	Summary	78
5.6	Controlled Sample: Humans and Models	78
5.6.1	Experiments and Results with Human Annotators	78
5.6.2	Comparison with humans’ errors	83
5.6.3	Summary	84
5.7	Discussion and Conclusion	85
5.8	Summary	88
6	Pre-trained Transformers Integrating Complementary Multimodal Information	89
6.1	Introduction	90
6.2	Related Work	92
6.3	Data	94
6.3.1	Data Collection	95
6.3.2	Meta-Annotation	97
6.4	Task	98
6.4.1	Dataset	100
6.5	Experiments	100

6.5.1	Models	101
6.5.2	Human Evaluation	102
6.6	Results	103
6.7	Analysis	104
6.8	Conclusion	107
6.9	Summary	107
7	Impact of Task Difficulty on Transfer Learning of Neural Multimodal Models	108
7.1	Introduction	108
7.2	Task Setup	110
7.3	Models and Experiments	111
7.4	Results and Analysis	113
7.5	Related Work	115
7.6	Conclusion	116
7.7	Summary	116
8	Conclusion	118
8.1	Contribution and Perspectives	119
9	Appendix	122
9.1	Chapter 3	122
9.2	Chapter 5	123
9.2.1	Game examples	123
9.3	Chapter 6	125
9.3.1	Further Details on Data (Sec. 3)	125
9.3.2	Further Details on Experiment (Sec. 5)	129
9.4	Chapter 7	131
9.4.1	Implementation details	131

9.4.2	Hyperparameter search	131
9.4.3	Continual Learning Evaluation Measures	131
9.4.4	Elastic Weight Consolidation	133
9.4.5	Confusion matrices	134
9.4.6	Neuron activations	134

Bibliography		134
---------------------	--	------------

List of Tables

3.1	Model comparison on the accuracy results for all games, and for those of 3/5/8 dialogue length.	41
3.2	Accuracy obtained on the test set containing dialogues in the Ground Truth order (GT) vs. the reversed order (Reversed).	42
3.3	Accuracy of the models when receiving all turns of the dialogue history and when removing the last turn (W/o last) or receiving only the last turn (Last) for dialogues with 3, 5, and 8 turns.	43
4.1	Question type distribution in successful games following the classification proposed in Shekhar et al. (2019) where a question can be assigned to more than one attribute type (multiple labels); the Single label column reports the number of questions which have been assigned to only one type.	54
4.2	Sub-type spatial questions distribution in successful games of questions annotated with only the spatial label in the test set (total: 29845).	55
4.3	Accuracy of the models on the successful games by question type based on the multi label assignment. Values in parenthesis report the comparison with LSTM.	56
4.4	Accuracy of the sub-type of spatial questions (successful games, questions assigned only one type)	57
4.5	Language to Vision attention in LXMERT: Number of regions of the image considered salient in the last layer from the CLS token – viz. regions with an attention value higher than the 0.05 threshold.	60
5.1	Statistics on the full test set and on the Controlled test set; both divided into the Yes- (resp. No-) subsets obtained by selecting only dialogues with a positively (resp. negatively) answered question in the last turn.	71

5.2	Full test set: Task Accuracy obtained by models when receiving: a) only the questions (Only Q); b) the full dialogue in the Yes-set vs. No-set, viz. games ending with a Yes-turn vs. a No-turn. All differences between RoBERTa and LXMERT are statistically significant.	75
5.3	Full test set: Accuracy comparison when giving to the model the dialogue without the last turn (W/o Last) or with only the last turn (Last). (The * marks RoBERTa’s and LXMERT’s scores whose differences are statistically not significant.)	75
5.4	Change across consecutive turns in the probability assigned to the target after Yes- vs. No- vs. N/A-turns, i.e., $P(o)_{T_{i+1}} - P(o)_{T_i}$ (full dialogue history in the full test set) and before/after the last turn (Last turn in games on which the model has succeeded).	77
5.5	Humans’ performance on controlled sample: percentage of games guessed correctly by at least two participants (MAJ) vs. by at least one participant (MIN). (* not significant)	81
5.6	Average time (seconds) taken by humans to solve games belonging to the different groups analyzed. Normalized with respect to the number of token in the text; only successful games are considered. (* not significant)	81
5.7	Controlled Sample. Removing turns: comparison of the task accuracy when models receive the full dialogue vs. only the No- vs. only the Yes-turns. Human accuracy computed with the majority vote. (*, ** not significant)	83
5.8	Controlled Sample. Without the last turn, Only the last turn, Full Dialogue: accuracy comparison to highlight the role of the last turn when it contains a positive (Yes-set) vs negative (No-set) answer.	83
5.9	Error Analysis: Percentage of games human failed among those failed by each model.	84
5.10	Error Analysis: Percentage of games in which each model does the same mistake made by humans (i.e., by selecting the same wrong candidate object as a human annotator).	85
6.1	Descriptive statistics of the dataset including, from left to right: 1) # (and %) of unique samples; 2) # of unique images; 3) # of unique intentions; 4) # of unique actions; 5) # of unique target actions; 6) # of unique decoy actions; 7) average number of tokens in intentions; 8) average number of tokens in actions.	99

6.2	Results for the 3 settings: L , V , and LV . s refers to transformer-based models trained from scratch. For each model, we report average accuracy and std over 3 runs. Human accuracy is computed over 300 samples (we report values based on both majority vote, i.e., 2 out of 3, and average of best participants; see 6.5.2).	103
6.3	Accuracy of the pretrained transformer-based models on the <i>hard</i> samples of the test set. Human accuracy is computed over 92 samples. . . .	106
7.1	Mean accuracy over 3 runs: Trained on each task independently (first two rows; per-task label space \mathcal{Y}) vs. CL setups (single-head label space over all \mathcal{Y}).	114
9.1	Epoch, validation accuracy, and number of parameters for best models. .	123
9.2	To each of the 54 clusters we have assigned a label that summarize its main topic as illustrated by the examples of the actions we report for each cluster. Each action has been annotated with codes to mark the verb (code1) or the complement object (code2) of the main sentence and the verb (code3) and the complements (code4) of the secondary sentence. The clusters are listed by their size (from the biggest to the smallest).	128
9.3	Statistics about the meta-annotation of the data. For each cluster, we report the number of actions, of verbs in the main sentence (code1) and in the secondary sentence (code3) and the number of nouns occurring as complements in the main sentence (code2) and in the secondary sentence (code4).	136
9.4	Examples of action with the word-type codes. Note that, (1) the same verb - e.g. <i>join</i> , line 1 and line 3 - in different clusters gets different codes; (2) the same object within the same cluster if in different syntactic positions (- e.g. <i>frisbee</i> in line 4), gets different codes but (3) the same object, in the same cluster, in the same syntactic position - e.g. <i>frisbee</i> , line 3 and line 4 - gets the same code.	137
9.5	Epoch and validation accuracy of the best models for each run.	137
9.6	Number of parameters of each model. The number of parameters is the same both in models trained from scratch and in pre-trained ones. . . .	137
9.7	Confusion matrix of the model trained independently on Wh-q.	138
9.8	Confusion matrix of the <i>Naive</i> model on the $WH \rightarrow Y/N$ setup.	138
9.9	Confusion matrix of the <i>Cumulative</i> model on the $WH \rightarrow Y/N$ setup. .	138
9.10	Confusion matrix of the best <i>Rehearsal</i> model on the $WH \rightarrow Y/N$ setup.	138

9.11	Confusion matrix of the best <i>EWC</i> model on the $W_H \rightarrow Y/N$ setup.	139
9.12	Confusion matrix of the model trained independently on $Y/N-q$.	139
9.13	Confusion matrix of the <i>Naive</i> model on the $Y/N \rightarrow W_H$ setup.	139
9.14	Confusion matrix of the <i>Cumulative</i> model on the $Y/N \rightarrow W_H$ setup.	139
9.15	Confusion matrix of the best <i>Rehearsal</i> model on the $Y/N \rightarrow W_H$ setup.	140
9.16	Confusion matrix of the best <i>EWC</i> model on the $Y/N \rightarrow W_H$ setup.	140

List of Figures

2.1	Architecture of a 2-layer feed-forward ANN.	8
2.2	Computation performed by an RNN.	10
2.3	A representation of the Transformer Encoder. Figure taken from the paper by Vaswani et al. (2017)	13
2.4	A diagram of a VGG-16 CNN architecture.	17
2.5	A representation of the ViT Transformer architecture. Figure taken from the paper by Dosovitskiy et al. (2020).	18
2.6	A representation of transfer learning. During pre-training (top), the network is pre-trained on a task (e.g., an image classification task trained on ImageNet, while during fine-tuning (bottom), the ANN is trained on a new task (e.g., an image classification task trained on Places (Zhou et al., 2017)). The weights of the layers in red can be either frozen or not. The final layer is replaced in order to accommodate the new task (e.g., having neurons for the classes of the new classification problem).	20
2.7	A representation of the process of integrating multimodal information coming from language and vision through product (top) or concatenation (bottom).	23
2.8	Representation of the LXMERT architecture. Figure taken from the paper by Tan and Bansal (2019).	27
2.9	Example of image and questions from CLEVR. Questions evaluate different visual reasoning skills. Figure taken from the paper by Johnson et al. (2017a).	29
2.10	Top: Statistics for CLEVR. Most questions are unique and few questions from the validation and test sets appear in the training set. Bottom left: Comparison of question lengths for different VQA datasets. Bottom right: Distribution of question types in CLEVR. Figure and analyses taken from the paper by Johnson et al. (2017a).	30

2.11	GuessWhat?! human dialogues are short and with a clear division of roles between players; most of the last questions are answered positively, are long, and contain details suitable to guess the target object.	30
2.12	Statistics of the training set (the validation and test sets have similar distributions). Dialogue length refers to the number of turns. Up: The distribution of Yes/No questions is very unbalanced across the clusters of games (the percentage of Yes answers is much higher in shorter dialogues); Middle In the large majority of games, the last question is answered positively; Bottom: The last questions are always longer (length of questions per turn for the clusters with dialogues having 3, 5, and 8 turns).	32
2.13	Up: longer human dialogues contain more distractors and more distractors of the same category of the target object, and more rare words; Down: The distribution of target objects is unbalanced, since “person” is the most frequent target.	33
3.1	Shared Encoder-Guesser skeleton. The Guesser receives the category labels (e.g., “bottle”) and the spatial coordinates (pos) of each candidate object. Multimodal encoders receive both the image and the dialogue history, whereas blind models receive only the latter.	39
3.2	Attention assigned by LXMERT-S to each turn in a dialogue when the dialogue history is given in the GT order (from QA1 to QA5) or in the reversed order (from QA5 to QA1).	44
3.3	A game solved successfully by LXMERT and not by RoBERTa (left) and a game solved by RoBERTa and not by LXMERT (right).	45
4.1	A vast amount of questions asked by humans in the GuessWhat?! game (de Vries et al., 2017a) are spatial. We classify them as <i>absolute</i> , <i>relational</i> , and <i>group</i> based on how they many objects are involved and how they are related. The red box marks the object(s) involved in the question, while the green box marks the target of the game. <i>Relational</i> and <i>group</i> questions need more than one object, whereas absolute do not.	48
4.2	Sample image and dialogue from the GuessWhat?! dataset. The red boxes mark the objects involved in the questions, while the green box marks the actual referent. LXMERT incorrectly answers “yes” to question 5. LXMERT, like all Oracles, does not have access to the dialogue history. It probably interprets the question as “is the target in the middle of the picture?”. The image and dialogue illustrates the history dependence of questions.	58

4.3	Attentions from the CLS: in absolute questions attention is mostly on the only object the question refers to (the left bus, 0.13) and the target object (0.64) (left); in the relational questions attentions spread between the two related objects (car and boat, 0.12 each) and the target object (the boat on the back, 0.9) (middle); in the group questions attentions goes to the entity of the referred group (0.08 and 0.13) and the target (0.37) (right).	60
5.1	Examples of dialogues from two asymmetric and partially observable visual dialogue data (PhotoBook and Meet Up! (Haber et al., 2019; Ilinykh et al., 2019a)) and a symmetric visual dialogue in which the answerer sees the image and the questioner does not see it (Das et al., 2017a; Chattopadhyay et al., 2017). For all datasets, we selected exchanges containing negation, the focus of our study.	65
5.2	Two samples of GuessWhat?! human dialogues ending with a positive (left) and a negative (right) turn.	67
5.3	Shared Encoder-Guesser skeleton. The Guesser receives the category labels (e.g., “bottle”) and the spatial coordinates (pos) of each candidate object. Multimodal encoders receive both the image and the dialogue history, whereas blind models receive only the latter.	72
5.4	Prolific interface: Humans were given a dialogue, an image with colored bounding boxes, and a numbered list of candidates with colors matching those of the bounding boxes. They had to use the keyboard device to choose the target.	79
5.5	Errors made by humans and computational models when receiving dialogues without the last turn.	85
6.1	One real sample of our proposed task. Given an image depicting, e.g., a tennis player during a match and the intention “ <i>If I have tons of energy</i> ”, the task involves choosing, from a list of 5 candidate actions , the target action that unequivocally applies to the combined multimodal input: “ <i>I will play a game of tennis with the man</i> ”. The task is challenging: a model exploiting a language or vision bias could fall into the trap of decoy actions containing words highlighted in blue or orange, respectively. Therefore, selecting the <i>target</i> action requires models perform a genuine integration of the two modalities, whose information is complementary. Best viewed in color.	91
6.2	Data collection. Examples of good (top) and bad (bottom) annotations provided to participants in the task instructions. Errors and corresponding warnings are shown to make participants familiarize with the tool.	95
6.3	Five $\langle intention, action \rangle$ tuples provided by 5 unique participants for the image in Figure 6.1.	96

6.4	Four samples from our dataset. I : Intention; T : Target action; L/V : Language-/Vision-based decoys.	99
6.5	Two samples where humans give the correct answer in the <i>LV</i> setting— but neither in <i>L</i> nor in <i>V</i> . LXMERT _{LV} picks the correct answer (blue) in the left sample, a wrong one (red) in the right sample. I : Intention; T : Target action; L/V : Language-/Vision-based decoys. Best viewed in color.	105
7.1	Overview of our linguistically-informed CL setup for VQA.	109
7.2	Analysis of the neuron activations on the penultimate hidden layer for the I) WH → Y/N setup. “equal_{shape,color,material,size}” refers to Y/N-q, “query_{..}” refers to WH-questions.	114
9.1	Example of game where RoBERTa correctly guesses the object, but LXMERT does not.	124
9.2	Example of game where LXMERT correctly guesses the object, but RoBERTa does not.	124
9.3	Example of game where models and humans are all wrong, but RoBERTa does the same mistake performed by one human.	125
9.4	Data collection. Examples of good/bad annotations provided to participants at the beginning of the task. Note that the errors and corresponding warnings are shown to make them familiarize with the tool.	125
9.5	Data collection. One annotation sample presented to participants. Given an image, participants are asked to provide an intention and an action. To ensure they are doing the task properly, a verification question is asked preliminarily. Answering the question correctly (multiple correct answers) leads to the annotation phase.	127
9.6	Analysis of the neuron activations on the penultimate hidden layer of the <i>Naive</i> model for the I) WH → Y/N setup.	135
9.7	Analysis of the neuron activations on the penultimate hidden layer of the model trained independently on Y/N-q.	135

Publications

This thesis collects results coming from different papers published during my PhD. In particular, most of the content of this thesis comes from the following publications:

- **Claudio Greco***, Alberto Testoni*, and Raffaella Bernardi. “Grounding Dialogue History: Strengths and Weaknesses of Pre-trained Transformers.” In International Conference of the Italian Association for Artificial Intelligence, pp. 263-279. Springer, Cham, 2020.
- Alberto Testoni, **Claudio Greco**, Tobias Bianchi, Mauricio Mazuecos, Agata Marcante, Luciana Benotti, and Raffaella Bernardi. “They are not all alike: answering different spatial questions requires different grounding strategies.” In Proceedings of the Third International Workshop on Spatial Language Understanding, pp. 29-38. 2020.
- Alberto Testoni*, **Claudio Greco***, and Raffaella Bernardi. “Artificial Intelligence models do not ground negation, humans do. GuessWhat?! dialogues as a case study.” *Frontiers in Big Data*. 2021.
- Sandro Pezzelle, **Claudio Greco**, Greta Gandolfi, Eleonora Gualdoni, and Raffaella Bernardi. “Be different to be better! A benchmark to leverage the complementarity of language and vision.” In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 2751-2767. 2020.

* These authors contributed equally to this work.

- **Claudio Greco**, Barbara Plank, Raquel Fernández, and Raffaella Bernardi. “Psycholinguistics Meets Continual Learning: Measuring Catastrophic Forgetting in Visual Question Answering.” In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3601-3605. 2019.

Chapter 1

Introduction

Humans are characterized by the ability to integrate information coming from multiple modalities, such as language and vision. Building computational models able to reproduce these abilities is the aim of multimodal learning. Researchers in this field have been working on tasks where models are required to rely on both language and vision (Baltrušaitis et al., 2018). These models are asked, for instance, to provide descriptions (Bernardi et al., 2016a), to answer questions (Sharma and Jalal, 2021), or to hold dialogues (Chen et al., 2020) about the visual content of images.

In the last years, pre-trained Transformers proved to work extremely well as computational models for tasks involving natural language processing (Devlin et al., 2018), computer vision (Dosovitskiy et al., 2020), and the interface between them (Li et al., 2019; Lu et al., 2019a; Tan and Bansal, 2019; Chen et al., 2019a; Su et al., 2020; and Nan Duan et al., 2020). Inspired by human ability to transfer knowledge acquired from previous tasks when learning new ones, these kinds of models are pre-trained on several different tasks in order to acquire useful knowledge which can be reused when learning from the downstream tasks. Likewise, inspired by human ability to focus on the most salient information, these kinds of models leverage an attention mechanism which allows them to focus on the most important parts of the input data.

Transformers seem to reach a high performance due to their reliance on pre-training and attention. However, we do not know whether and how well they exploit these fea-

tures to solve more complex tasks such as answering questions that require encoding relations among regions, or answering questions that require both linguistics and visual grounding. Our research aims to investigate to what extent pre-trained multimodal Transformers effectively rely on pre-training and on attention mechanisms when dealing with multimodal tasks which require the association of several information cues to be solved. We are particularly interested in testing their ability to:

- Detect salient information in each modality while grounding one into the other (Chapters 3, 4, and 5)
- Combine complementary information provided by the two modalities (Chapter 7)

We tackle the first issue by taking the dialogues of a visual guessing game, namely *GuessWhat?!*, as case study. The structure of the dialogues in this task is suitable for our purpose since it is rather simple (a short sequence of polar questions and their respective answers referring to an object in an image) and hence easy to manipulate when necessary. In particular, pre-trained Transformers are required to:

- Select the object in the image the dialogue refers to in the referential guessing task. We focus on understanding whether models are able to detect the most salient information in the dialogue history and to properly integrate the answer and the corresponding question (Chapters 3 and 5)
- Answer questions about an object in the image. We focus on spatial questions since they challenge models on detecting information across regions (Chapter 4)

We tackle the second issue by testing pre-trained Transformers on an ad hoc built task which consists in guessing the most plausible action based on an utterance and an associated image conveying different but complementary information. In all the investigations mentioned above, we compare pre-trained Transformers with their counterparts trained from scratch and architectures not relying neither on pre-training nor on attention.

In our research, we are also interested in the crucial issue of catastrophic forgetting, which affects both classical computational models and pre-trained Transformers. As

pointed out by studies in the continual learning field (Ring, 1997), models should be not only able to transfer their knowledge to new tasks, but also to avoid forgetting how to solve previously-solved tasks after having learned a new task. We tackle this parallel issue by investigating if a better control of tasks' order helps to mitigate catastrophic forgetting in visual question answering tasks involving polar and Wh questions about images (Chapter 7). In this case, we use classical multimodal models instead of pre-trained Transformers since we consider that obtaining results on relatively simple architectures are needed to preliminary deal with this general issue.

Chapter 2

Background

In the last years, the advent of deep learning models based on artificial neural networks has brought to steep advances in the field of natural language processing and computer vision. Moreover, the advent of architectures transferring previously-acquired knowledge on several tasks has been a breakthrough in natural language processing and computer vision. However, models learning purely from textual data lacked the processing of the perceptual information used by humans. Hence, many models and tasks have been proposed in order to evaluate the capability of models to integrate information coming from modalities such as language and vision. In this chapter, first we describe how artificial neural networks generally work. Then, we focus on state-of-the-art models for natural language processing and computer vision. Subsequently, we describe how transfer learning has been employed to build pre-trained models which can be easily adapted in order to solve the task at hand. Finally, we focus on multimodal models, tasks and datasets designed in order to integrate the linguistic and visual modalities.

2.1 Artificial Neural Networks

The goal of an Artificial Neural Network (ANN) is to approximate a mathematical function in order to obtain a certain output starting from the given input ([Goodfellow et al., 2016](#)). For instance, the mathematical function to approximate may correspond to a

textual classifier labeling a given e-mail as spam or not-spam or to a visual classifier labeling a given picture as depicting a dog or a cat. The mapping is defined by learning the parameters of the ANN leading to the best approximation of the function.

Feed-forward ANNs are called feed-forward because there are no feedback connections in which some outputs of the model are fed back into itself. In particular, the input flows through several intermediate computation layers and finally goes to the output layer which computes the output. The first layer is called the input layer, while the last layer is called the output layer. The middle layers are called hidden layers, because their values are not observed in the training set. Each layer contains a number of neurons, which are computational units having scalar inputs and outputs. Each input has an associated weight. Each neuron multiplies each input by its weight, sums the weighted inputs, and eventually applies a non-linear function to the result. The computation of the output value of a neuron is called activation. The neurons of a layer are connected to the neurons of the next layer, composing the architecture of the ANN.

Given an ANN composed of n_l layers l_1, l_2, \dots, l_{n_l} , $\mathbf{W}_{ij}^{(l)}$ denotes the weight of the connection from the j -th neuron in layer l and the i -th neuron in layer $l+1$. The simplest ANN is the perceptron, which is a linear function of its inputs defined as follows:

$$f(x) = \mathbf{W}\mathbf{x} + \mathbf{b},$$

where $\mathbf{W} \in \mathbb{R}^{d_{out} \times d_{in}}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^{d_{out}}$ is the bias term, and $\mathbf{x} \in \mathbb{R}^{d_{in}}$ is the input vector which represents the data given as input to the ANN.

In order to go beyond linear functions, a hidden layer having a non-linear function is introduced, resulting in a 2-layer feed-forward ANN defined as follows:

$$f(x) = \mathbf{W}^{(2)}g(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)},$$

where $\mathbf{W}^{(1)} \in \mathbb{R}^{d_1 \times d_{in}}$ and $\mathbf{b}^{(1)} \in d_1$ are the weight matrix and bias term for the first layer, $\mathbf{W}^{(2)} \in \mathbb{R}^{d_2 \times d_1}$ and $\mathbf{b}^{(2)} \in d_2$ are the weight matrix and bias term for the second layer, g is a non-linear function, also called non-linearity or activation function, which is applied element-wise to its input vector, and $\mathbf{x} \in \mathbb{R}^{d_{in}}$ is the input vector of the ANN.

Figure 2.1 shows the architecture of a 2-layer feed-forward ANN. Deeper ANNs, which approximate more complex functions, can be built by adding more layers. In general, ANNs having at least one hidden layer are called Multi-Layer Perceptrons (MLPs).

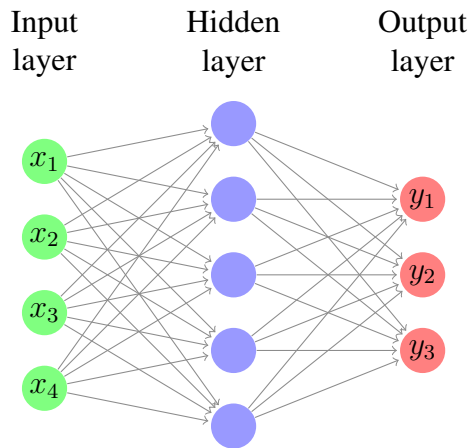


Figure 2.1: Architecture of a 2-layer feed-forward ANN.

Supervised learning is a common approach to find the parameters of an ANN which best approximate a given function based on a training set of input-output pairs. This paradigm requires to compute a loss function which represents, for each input-output pair in the training set, the error committed by the ANN when computing the output for input of the pair with respect to the actual output. In particular, the loss function can be defined as $\mathbb{E}[L(f(x; \Theta), y)]$, where Θ is the matrix containing parameters, $f(x; \Theta)$ is the approximated function, (x, y) is a input-output pair sampled from the training set, and \mathbb{E} averages the loss over samplings. The parameters of an ANN are often randomly sampled according to a given probability distribution. Then, the back-propagation algorithm computes the gradient $\frac{\partial L}{\partial w}$ of a loss function L representing the error committed by the ANN with respect to any parameter w (Rumelhart et al., 1988). Finally, the parameters of the ANN minimizing the loss function are adjusted according to a gradient-based optimization method (LeCun et al., 2012) such as Adam (Kingma and Ba, 2014).

2.1.1 Natural Language Processing

As mentioned previously, an ANN can be used to build a classifier labeling a given e-mail as spam or not-spam. In order to do that, the ANN must build a representation

of the received textual data, where the representation corresponds to the output vector computed by a layer which is given as input to the final layer computing the actual classification. In natural language processing, a simple way to deal with a sequence of words consists in representing it as the average of its word representations computed through methods such as Word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Then, more powerful methods, which can build a richer representation from the received textual data starting from its word representations (having parameters initialized randomly or through methods like Word2Vec), have been developed. Initially, Recurrent Neural Networks (Rumelhart et al., 1985), which represent the input data sequentially by updating a state based on the next element of the sequence and the last previously-updated state, have been proposed. Subsequently, Vaswani et al. (2017) proposed Transformers, which receive the sequence all at once and leverage an attention mechanism to simultaneously focus on each part of the sequence given each other part of the sequence. These last two approaches are described as follows.

Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a class of ANNs where connections between neurons form a directed cycle (Rumelhart et al., 1985). This allows the RNN to process the input sequentially, by updating a state based on the next element of the sequence, with the output depending on the previous computations. The state after the processing of the whole sequence is usually taken as the representation of the input data.

RNNs are described as operating on a sequence which contains vectors $\mathbf{x}(t)$ with the time step index t ranging from 1 to τ . According to the idea of defining recurrent connections between neurons, different RNN architectures can be designed. In general, given the input vector $\mathbf{x}^{(t)}$ at time-step t , a variable \mathbf{h} representing the state of the hidden units and the weights θ , RNNs compute the values of their hidden units as follows:

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \theta).$$

Usually, the hidden state vector $\mathbf{h}^{(t)}$ is initialized to a fixed vector at time step 0. An

RNN learns to use $\mathbf{h}^{(t)}$ as a summary of the relevant aspects of the past sequence of inputs up to t . This summary is lossy because it maps an arbitrary length sequence $(\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(1)})$ to a fixed length vector $\mathbf{h}^{(t)}$. Ideally, an RNN should be able to learn to keep the most important aspects of the input sequence to provide a representation rich enough to solve the task at hand. Figure 2.2 shows a representation of the computation of an RNN. In particular, on the left it is shown a representation of an RNN, while on the right it is shown its unfolding, that is its computation on each element of the sequence.

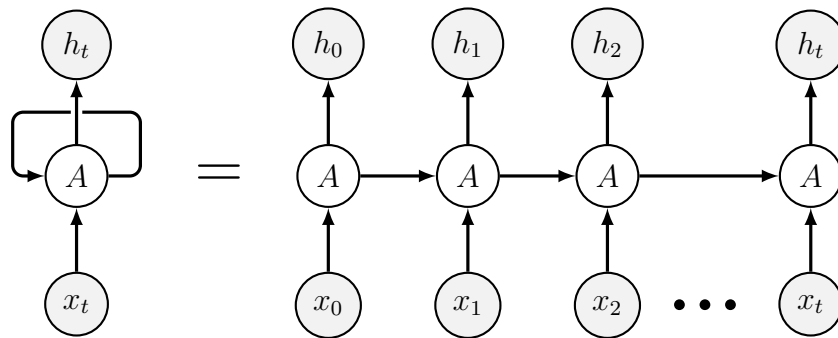


Figure 2.2: Computation performed by an RNN.

Typical RNNs present output layers which exploit the information contained in $\mathbf{h}^{(t)}$ in order to make predictions. Given the input vector $\mathbf{x}^{(t)}$, the weight matrices \mathbf{U} , \mathbf{V} and \mathbf{W} representing respectively the weights of input-to-hidden, hidden-to-output and hidden-to-hidden connections, the bias terms \mathbf{b} and \mathbf{c} , the hidden state vector $\mathbf{h}^{(t)}$, and the non-linearities f and g , the output of a typical RNN is computed as follows:

$$\begin{aligned}
 \mathbf{a}^{(t)} &= \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} + \mathbf{b} \\
 \mathbf{h}^{(t)} &= f(\mathbf{a}^{(t)}) \\
 \mathbf{o}^{(t)} &= \mathbf{V}\mathbf{h}^{(t)} + \mathbf{c} \\
 \hat{\mathbf{y}}^{(t)} &= g(\mathbf{o}^{(t)}).
 \end{aligned}
 \tag{2.1}$$

The output of an RNN is the composition of several number of non-linear transformations. Unfortunately, even if each of these transformations is smooth, their composition might not be. In particular, the derivatives through the whole composition tend to be either very small or very large. This issue, called vanishing/exploding gradient (Bengio et al., 1994), prevents the RNN to reach a good minimum of the loss function. Due to

this issue, RNNs are unable to learn long-term dependencies very well.

In order to effectively learn long-term dependencies in a sequence, many extensions to RNNs have been proposed, such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) units and Gated Recurrent Units (GRUs) (Chung et al., 2014), which selectively add or remove information to their state according to particular structures called gates. Given the input vector $\mathbf{x}^{(t)}$, the weight matrices \mathbf{W}_f , \mathbf{W}_i and \mathbf{W}_o representing respectively the weights of forget, input and output gates, the bias terms \mathbf{b}_f , \mathbf{b}_i , and \mathbf{b}_o representing respectively the bias terms of forget, input and output gates and the cell state vector \mathbf{c}_t , the output of an LSTM is computed as follows:

$$\begin{aligned} \mathbf{f}_t &= \text{sigmoid}(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \\ \mathbf{i}_t &= \text{sigmoid}(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \\ \mathbf{o}_t &= \text{sigmoid}(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \\ \tilde{\mathbf{c}}_t &= \text{htan}(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \\ \mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t \\ \mathbf{h}_t &= \mathbf{o}_t \circ \text{htan}(\mathbf{c}_t), \end{aligned}$$

RNN-based models have been used extensively to model language sequentially, e.g. processing words while reading them from the left to the right or vice-versa. Bidirectional models have then be proposed in order to capture dependencies across the words in a sequence in a more effective manner (Schuster and Paliwal, 1997). These kind of models work as follows. First, they encode each word reading the sequence from the left to the right. Then, they do the opposite, encoding each word reading the sequence from the right to the left. Finally, they generate the final representation of each word by concatenating the representations produced starting from both sides of the sequences.

Transformers

The Transformer (Vaswani et al., 2017) is a class of ANNs which performed exceptionally well in several natural language processing tasks, such as summarization and

machine translation. Previously, RNNs were the standard approach to process sequential data such as natural language. Bahdanau et al. (2014) further improved RNNs by equipping them with an attention mechanism, which allows the model to focus on the most relevant parts of the sequence. Transformers remove the sequential processing introduced by RNNs and instead leverage only a particular query-key-value attention mechanism to simultaneously focus on the most important parts of the sequence. By processing data in parallel, Transformers provide a higher computational efficiency and scalability than RNNs, making it possible to train models of unprecedented size.

The Transformer has been developed according to the Sequence-To-Sequence framework (Sutskever et al., 2014a) (also called Encoder-Decoder framework (Cho et al., 2014)), which transforms a given sequence of elements, such as the sequence of words forming a sentence, into another sequence. The Sequence-To-Sequence framework consists of an Encoder, which represents the received input as a vector, and a Decoder, which generates the desired output from the input representation. The Encoder-Decoder framework has become popular in the field of machine translation. In that context, for instance, the Encoder represents the English sentence "The weather is nice" and the Decoder receives its representation in order to generate its French translation "Il fait beau". This thesis will focus on the encoder, which has the important role of summarizing the most relevant aspects of the input in order to obtain a representation for the decoder. Hence, when describing Transformers, we will focus on the explanation of Encoder.

Figure 2.3 shows a representation of the Encoder of a Transformer. Each block of the Encoder consists in a stack of identical blocks, each composed of a self-attention and a feed-forward layer. Transformers employ a residual connection around each of the two sub-layers, followed by layer normalization (He et al., 2016). That is, the output of each sub-layer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is the function implemented by the sub-layer itself. Each token of the input sentence is encoded into a vector using an embedding algorithm. Since Transformers do not process input sequentially, the token representations must contain information about their position. Hence, they leverage a particular embedding algorithm to generate word representations having a positional encoding (Gehring et al., 2017). These vectors first flow through a

self-attention layer, which allows the Encoder to look at the other representations in the sequence when encoding each token. The output of the self-attention layer is then processed by a feed-forward ANN, which is applied to each token representation. After being processed by the self-attention layer and the feed-forward layer, the representation is given as input to the next block which processes it as in the previous one.

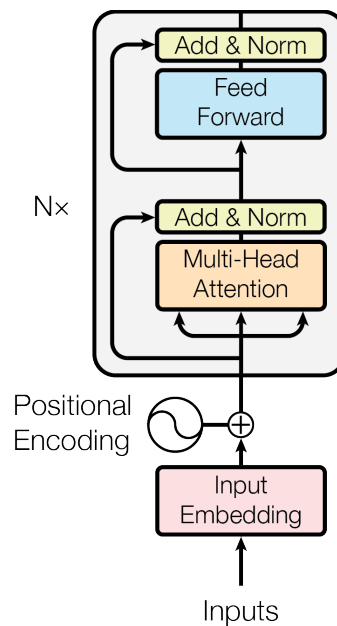


Figure 2.3: A representation of the Transformer Encoder. Figure taken from the paper by Vaswani et al. (2017)

We describe, as follows, the self-attention mechanism exploited in Transformers. The basic attention mechanism is called Scaled Dot-Product Attention. This building block, which allows Transformers to weight each token of the sequence differently when reading each token, is leveraged in the Multi-head Attention mechanism to focus on tokens according to several dimensions (e.g., subject-object relationships or sequence length).

Scaled Dot-Product Attention This attention mechanism allows the network to focus on the most salient tokens of the sequence when encoding each token. Scaled Dot-Product Attention is computed as follows. First, the embedding of each token is multiplied by three matrices whose weights are learned during the training process in order to generate a query, a key, and a value vector for each token. Then, the network computes a score for each token, determining how much focus to place on each part

of the sentence as we encode a given token. The score is computed by taking the dot-product of the query vector of the token we are computing the representation of with the key vector of each token in the sequence. These scores are divided by $\sqrt{d_k}$. A Softmax operation is applied to the scores in order to obtain positive scores which add up to 1. The value vector of each token is then multiplied by the score for that token. Finally, all the resulting weighted value vectors are summed in order to compute the output of the self-attention layer for the given token. In practice, query, key, and value vectors are packed together into matrices Q , K , and V , respectively, in order to compute them simultaneously. Formally, Scaled Dot-Product Attention can be defined as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.2)$$

Multi-head Attention Instead of performing a single attention mechanism, Transformers linearly project the queries, keys, and values h times with different linear projections whose weights are learned during the training process. Then, the attention mechanism is performed in parallel on each of the projected versions of the query, key, and value vectors. Finally, the outputs of the computed attention mechanisms are concatenated and projected again using another matrix having learned weights in order to obtain the final values. This allows the model to focus on several dimensions of the same parts of the sequence. Formally, Multi-head attention can be defined as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.3)$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

2.1.2 Computer Vision

As mentioned previously, an ANN can be used to build a classifier labeling a given picture as depicting a dog or a cat. In order to do that, the ANN must build a representation of the received visual data, where the representation corresponds to the output

vector computed by a layer which is given as input to the final layer computing the actual classification. The first approaches to build image classifiers were based on ANNs. The images received as input were usually represented by vectors containing their gray-scale pixel intensities. Subsequently, Convolutional Neural Networks have proven to be especially effective for building rich representations of visual data. Lately, more powerful methods, which extract region-level features and leverage attention to reason about them, have been proposed. These last two approaches are described as follows.

Convolutional Neural Networks

Loosely inspired by human visual perception, Convolutional Neural Networks (CNNs) are feed-forward ANNs which leverage particular structures called kernels to represent different perceptors responding to various stimuli in the given data (LeCun et al., 1995). CNNs are often faster and easier to be trained than classical ANNs since they do not connect each neuron of a layer with each other neuron of another layer. Moreover, they have been proven to be invariant to translation of the input data.

CNNs are especially suitable for processing images. Whereas in feed-forward ANNs the pixels of a given image were usually flattened in a single vector containing their gray-scale pixel intensities, CNNs receive a given image taking all the channels representing their visual content into account (usually the red, green, and blue channels) representing the pixels as a tensor having shape (image width, image height, 3). CNNs are usually composed of several kind of layers, which are described as follows.

Fully-connected layer This is the basic kind of layer employed in feed-forward ANNs. A fully-connected layer is composed of several neurons, each connected to each other neuron of the previous layer through a weighted connection. Basically, this kind of layer applies a linear transformation to the input vector through a weights matrix.

Convolutional layer This kind of layers relies on a specific feature detector, called filter or kernel, which moves across the receptive fields of the image checking for the

presence of a particular kind of (learned) feature in each receptive field. Hence, unlike fully-connected layers, convolutional layers do not have neurons connected to each other neuron of the previous layer. The feature detector is a two-dimensional array of weights. The kernel is applied to an area of the image, and a dot product is calculated between its weights and the input data. This dot product is then fed into an output array. Then, the kernel shifts by a certain fixed number of elements, called stride, repeating the process until it has moved across the entire image. The output of a convolutional layer is a feature map containing the result of the several dot products performed by the kernel across the input data. The feature map generation process is known as convolution.

Pooling layer Similar to the convolutional layer, the pooling layer reduces the spatial size of the input data. Unlike a convolutional layer, it does not add any additional parameter to the network. This improves the efficiency of the network by reducing the size of the input and promotes the emergence of higher level features. There are two types of pooling: max pooling and average pooling, which return the maximum value and the average value, respectively, of the portion of data covered by the kernel.

Several architectures of CNNs have been proposed. Examples of these architectures are LeNet-5 (LeCun et al., 1998), VGGNet (Simonyan and Zisserman, 2014), and ResNet (He et al., 2016). Figure 2.4 represents the architecture of a VGGNet designed for an image classification task. First, VGGNet applies several convolutional and max pooling layers applied one after another. Then, it flattens the output of the last max pooling layer into a single vector to which it applies several fully-connected layers having a ReLU activation function. Finally, it employs a softmax function to obtain a probability distribution over the possible classes of the classification problem.

Region-based Neural Networks

Convolutional Neural Networks have proven to be very effective in order to process images, but they might not be very effective in taking fine-grained information into account. Hence, methods extracting region-level features from images and employing an attention mechanism to reason about them have been developed.

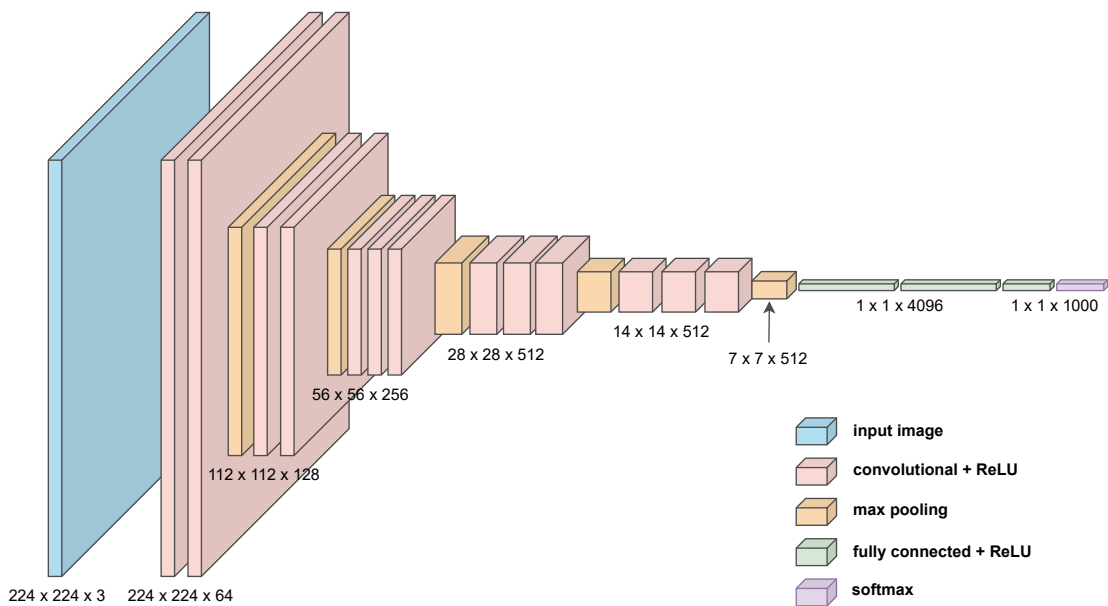


Figure 2.4: A diagram of a VGG-16 CNN architecture.

Anderson et al. (2018a) proposed a two-stage attention mechanism which splits the attention mechanism into two steps: bottom-up attention and top-down attention. The former uses an object detection model in order to detect the most salient regions in an image, whereas the latter attends to the most relevant regions detected by the former mechanism during the generation of the next word in the description. The two stage attention mechanism proposed in Anderson et al. (2018a) leverages a Faster R-CNN trained on Visual Genome in order to detect the most salient regions. This kind of approach obtained a really good performance, but was not able not capture the relationships which occurs between the different regions in an image.

Transformers, originally proposed in the context of natural language processing, have also been adapted to image classification. For instance, ViT (Dosovitskiy et al., 2020) is a model based as closely as possible on the original Transformer architecture. Similarly to what happens in traditional Transformers, where a sentence is represented as a sequence of word embeddings, ViT represents an input image as a sequence of image patches. In particular, it splits an image into a grid of square patches. Each patch is

then flattened into a single vector containing the channels of all the pixels in the patch. The resulting vector is then projected to a desired input dimension. Finally, A learned position embedding is added to each patch in order to allow the model to learn about the spatial structure of images. The attention mechanism employed in the Transformer architecture automatically allows the model to focus on the most important regions given each region of the image. When trained on a sufficient amount of data, ViT outperformed state-of-the-art CNNs with fewer computational resources. A representation of the architecture is shown in figure 2.5, which shows the whole process of extracting patches from the image, projecting them, concatenating them with position embeddings, and finally giving the resulting concatenation as input to the Transformer Encoder in order to obtain a representation which is given as input to an MLP, which generates the final distribution over the possible classes of the classification problem.

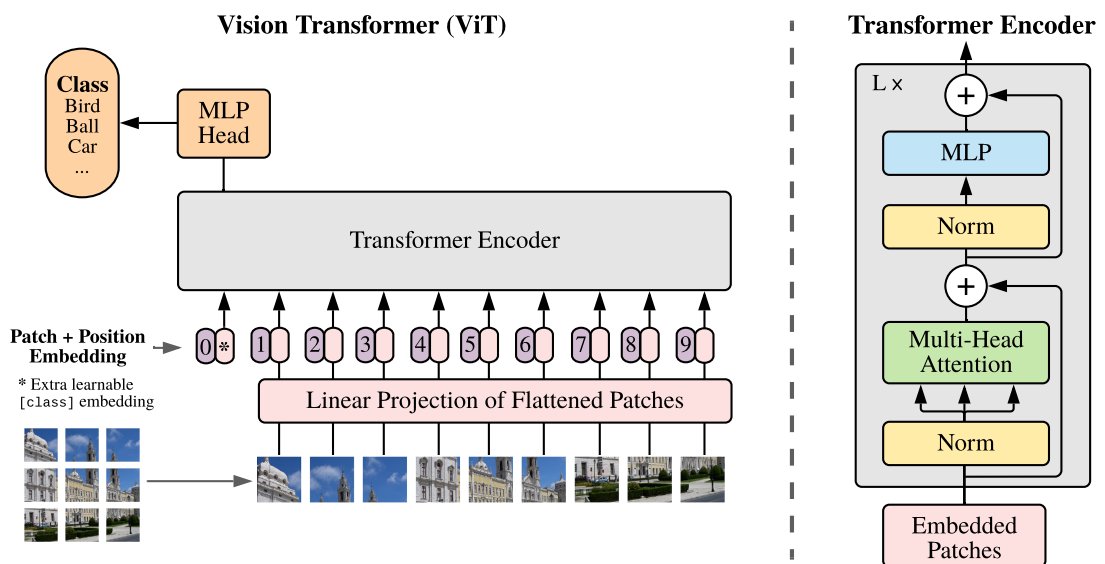


Figure 2.5: A representation of the ViT Transformer architecture. Figure taken from the paper by [Dosovitskiy et al. \(2020\)](#).

2.2 Transfer Learning

When ANNs are trained from scratch, their weights are initialized randomly and the values for the weights which best approximate the target function are computed through the training algorithm from the data at hand. Through the learning process, the out-

put layers of ANNs learn how to properly represent the input data in order to finally generate the desired output from the given input. However, representations (i.e., output values produced by layers) learned when representing input images in order to recognize bicycles could also apply when trying to recognize motorcycles. This led to transfer learning, which focuses on reusing representations learned while solving one problem on a different, but related, problem. In ANNs, transfer learning is usually achieved as follows. First, an ANN is trained on a task, e.g. on classifying pictures as bicycles or not bicycles. Then, task-specific layers are replaced with new randomly-initialized layers suitable for the new task. In a simple ANN classifying images, the task specific layer is usually the last one, where each neuron represents one possible class. Finally, the resulting architecture is trained on the new task, e.g. on classifying pictures as motorcycles or not motorcycles. The process of taking an ANN pre-trained on a task and training it on a new task starting from the previously-learned weights is known as fine-tuning. Figure 2.6 represents transfer learning, depicting the pre-training (top) of an ANN for image classification and the fine-tuning (bottom) of the same ANN on a different image classification problem. The weights of the layers in green can be either frozen or not. In the latter case, they are adapted in order to suit better the new task. The final layer is replaced in order to accommodate the new task (e.g., having neurons for the classes of the new classification problem).

Sometimes, an ANN is also trained in order to approximate two target functions at the same time. This is done by training the ANN through a loss function which takes into account the error on several tasks at the same time, e.g. on classifying images as cars or not cars and as bicycles or not bicycles. This process, which is known as multi-task training, allows the ANN to learn more general and richer representations. Then, the ANN can be fine-tuned on another task as explained above starting from these more general representation previously learned through multi-task learning.

2.2.1 Natural Language Processing

In the context of Natural natural language processing, transfer learning has been originally employed as follows. An embedding algorithm was used in order to generate

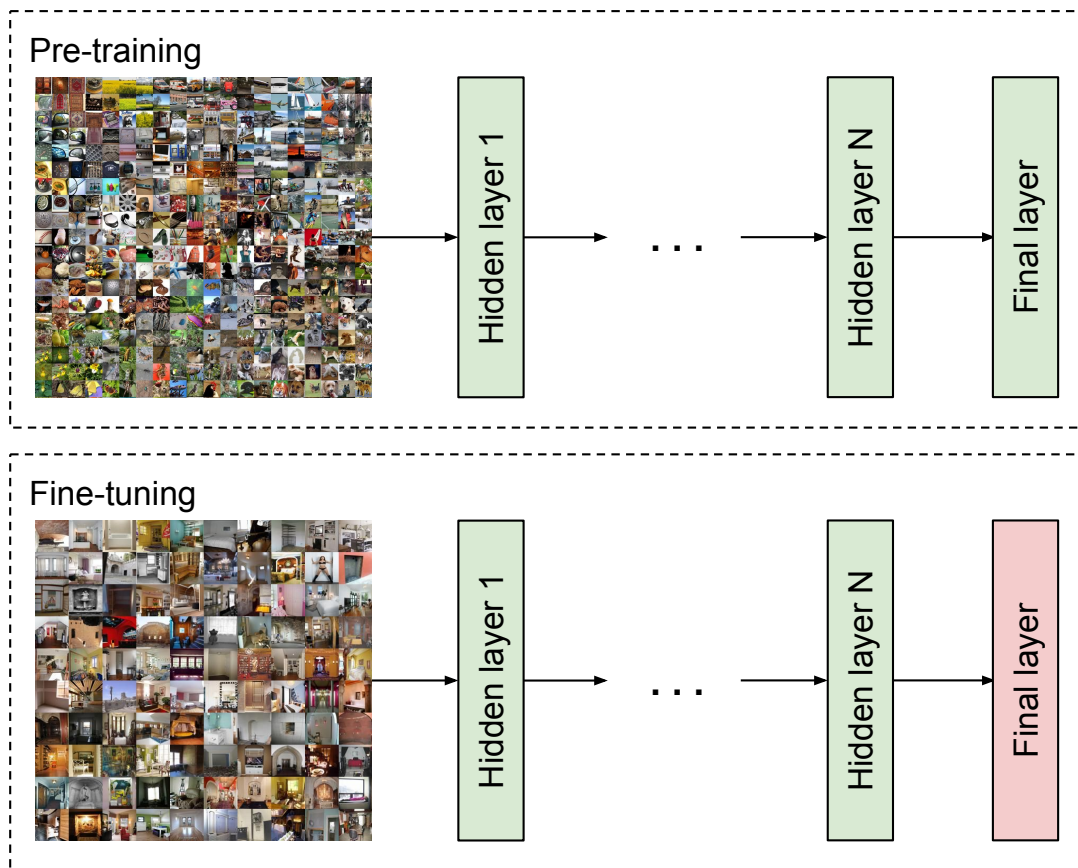


Figure 2.6: A representation of transfer learning. During pre-training (top), the network is pre-trained on a task (e.g., an image classification task trained on ImageNet, while during fine-tuning (bottom), the ANN is trained on a new task (e.g., an image classification task trained on Places (Zhou et al., 2017)). The weights of the layers in red can be either frozen or not. The final layer is replaced in order to accommodate the new task (e.g., having neurons for the classes of the new classification problem).

word embeddings such as Word2Vec or GloVe from a large corpus. Then, the sequence of word embeddings corresponding to the words of the input sequence was given as input to the RNN. The weights associated to word embeddings could be either fixed or fine-tuned during the training of the RNN in order to be adapted to the target task.

In the last years, Transformers have proven to be really suitable architectures to be pre-trained on language modeling tasks and then fine-tuned to the target task. BERT (Bidirectional Encoder Representations from Transformers) is a Transformer-based model which presented state-of-the-art results in a wide variety of natural language processing tasks (Devlin et al., 2018). The high performance of pre-trained Transformers led to a breakthrough in transfer learning and natural language processing. Basically, BERT

is Transformer pre-trained with multi-task training on two tasks involving bidirectional training. This is in stark contrast with the previous efforts in language modeling which usually looked at the input sequence either left-to-right or right-to-left or which combined the two training directions after having performed them independently. Indeed, BERT shows that a language model trained in a bidirectional way can have a deeper understanding of the context in the input sequence. In particular, the model is pre-trained on two tasks: Masked Language Modeling and Next Sentence Prediction. In the former, some percentage of the input tokens is masked randomly, and the model must predict those masked tokens. In the latter, when choosing the sentences A and B for each pre-training example, 50% of the time sentence B is the actual next sentence that follows sentence A (predicting the label *IsNext*), and 50% of the time it is a random sentence from the corpus (predicting the label *NotNext*). The model must predict whether sentence B follows sentence A (predicting the label *isNext*) or vice-versa (*NotNext*). BERT is pre-trained on a large corpus composed of sentences taken from BooksCorpus (800M words) (Zhu et al., 2015a) and English Wikipedia (2,500M words).

2.2.2 Computer Vision

In the context of Computer vision, the pre-training of CNNs on huge datasets such as ImageNet (Deng et al., 2009) led to a breakthrough in transfer learning, since it brought to several state-of-the-art results in image classification (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2016) and to pre-trained architectures which have been fine-tuned also for a wide variety of different tasks. Pre-trained CNNs are usually employed in two different ways:

Frozen feature extractor All the weights of the pre-trained CNN are frozen, since the CNN is used just as feature extractor. The representations computed by the CNN are given as input to a layer (e.g. a classifier) which re-purposes them for the new target task.

Fine-tuning of the whole model A few of the top layers of the CNN are unfrozen and the fine-tuning jointly adapts both those layers and the newly added layer (e.g. a

classifier). This allows the CNN to adapt the computed representations to the new target task.

Analogously to what happened in natural language processing, transfer learning has been employed also in Transformer-based architectures for Computer Vision. For instance, the previously-described ViT model has been pre-trained on ImageNet in order to provide a pre-trained model which can be fine-tuned on the task at hand. Through fine-tuning on an image classification task, the pre-trained ViT model performs really well compared to state-of-the-art approaches based on CNNs.

2.3 Multimodal Learning

Language and vision are two important modalities which humans combine a lot to understand their surroundings. However, models learning purely from textual data lacked the processing of the rich perceptual information used by humans. Hence, many models and tasks have been proposed in order to evaluate the capability of mixing information coming from modalities such as language and vision. In this section, we describe, first, the computational approaches leveraged in order to integrate language and vision and, then, the main multimodal tasks employed in the following chapters.

Many approaches have been proposed in order to integrate information coming from language and vision. The first methods were based on the idea of putting the representations coming from the two modalities together in single representations which were given as input to some layers meant to find patterns in the received multimodal data. Then, research in multimodal learning focused on attention-based methods, building architectures which automatically focus on the most salient parts of the two modalities when integrating them. These two approaches are defined as follows.

2.3.1 Fusion by product or concatenation

Given the representation $r_L = ENC_L(L)$ of linguistic data L through encoder ENC_L and the representation $r_V = ENC_V(V)$ of visual data V through encoder ENC_V , the integration can be performed either by product or concatenation. In the former case, $r_{LV} = r_L \cdot r_V$, i.e., the multimodal representation is obtained by multiplying the linguistic representation r_L and the visual representation r_V through an element-wise multiplication. In the latter case, $r_{LV} = [r_L, r_V]$, where $[a, b]$ represents the operator concatenating representations a and b , i.e., the multimodal representation is obtained by concatenating the linguistic representation r_L and the visual representation r_V . Figure 2.7 shows a representation of the process of integrating language and vision through product (top) or concatenation (bottom). The multimodal representation r_{LV} can then be given as input to some layers in order to perform the multimodal task at hand.

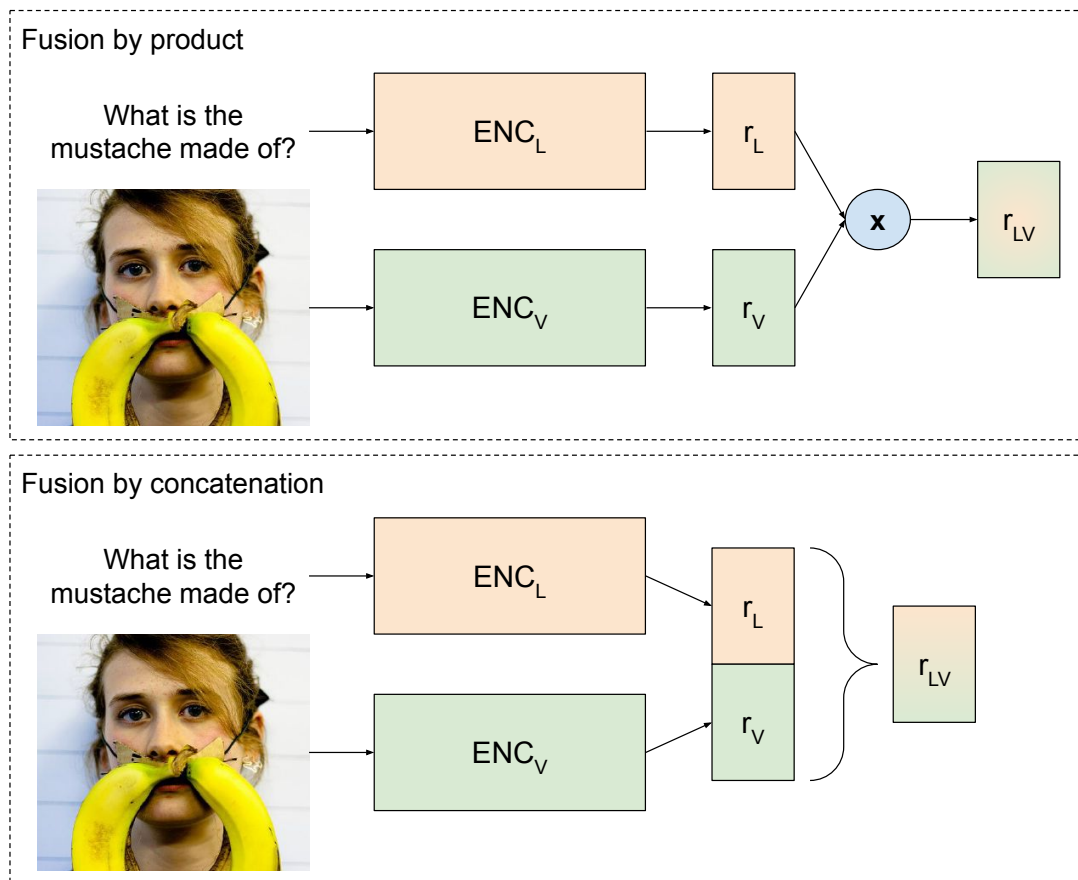


Figure 2.7: A representation of the process of integrating multimodal information coming from language and vision through product (top) or concatenation (bottom).

2.3.2 Fusion by cross-modal attention

Integrating information coming from different modalities requires being able to focus on several pieces of information and to find meaningful patterns between them. As it happened in natural language processing and computer vision, attention-based mechanisms have been employed also in multimodal models in order to provide a more fine-grained integration of the information coming from language and vision.

Several kinds of attention-based multimodal mechanisms have been proposed. In general, single-stage attention mechanisms take as input linguistic and visual representations and they learn how to focus on the most salient parts of the image encoded as a whole, without encoding the single regions independently. Two-stage attention mechanisms, instead, split the attention mechanism into two steps: bottom-up attention and top-down attention. The former uses an object detection model to detect the most salient regions in an image, whereas the latter attends to the most relevant regions detected by the former mechanism. Transformers have also been employed in order to focus on the most salient words and regions simultaneously through their attention mechanism.

With regards to single-stage attention mechanisms, [Xu et al. \(2015\)](#) proposed an approach which allowed the model to guide the generation of the description of an image focusing on the most salient features of its encoding. In particular, when generating the next word of the description, the authors tried both a "hard" approach which identifies the most important part of the image to focus on and a "soft" approach which weights each part of the image differently taking the encoding of multiple pieces of information into account at the same time. Since not all the words in a caption correspond to an image region, [Lu et al. \(2017\)](#) proposed a mechanism which decides when to rely on image regions during the generation. Similar single-stage attention mechanisms have been employed also in order to build models able to answer questions about images ([Fukui et al., 2016](#); [Shih et al., 2016](#); [Yang et al., 2016](#); [Xu and Saenko, 2016](#)) or to hold a dialogue about images ([Das et al., 2017b](#); [Guo et al., 2019](#)).

Splitting the attention process in two stages, the detection of image regions first and the actual attention on the previously-detected regions, the two-stage approach proposed by

Anderson et al. (2018a) obtained a very high performance on the 2017 VQA challenge, whose main goal was to build models able to answer questions about images. Yao et al. (2018) proposed a similar approach in order to generate a description of the content of an image. In particular, the authors built a graph over the detected objects in an image based on their spatial and semantic connections and leveraged an attention mechanism to reason about the objects when generating a description of the content of the image.

The last years have seen an increasing popularity of models which leverage the attention mechanism used in Transformers to focus over image regions and text at the same time. Inspired by BERT, these models are usually trained on several tasks to reach task-agnostic multimodal representations (Li et al., 2019; Lu et al., 2019a; Tan and Bansal, 2019; Chen et al., 2019a; Su et al., 2020; and Nan Duan et al., 2020). Lu et al. (2019b) proposed ViLBERT, a model which exploits the attention mechanism used in Transformers to focus over image regions and text segments. In particular, this model employs the self-attention mechanism over regions and over question tokens independently and uses co-attentional transformer layers to enable the exchange of information between modalities. ViLBERT has been extended through multi-task training involving 12 datasets (Lu et al., 2020). LXMERT is another multimodal Transformers which differs from ViLBERT by employing more multimodal pre-training tasks in order to learn more complex relationships between images and text (Tan and Bansal, 2019).

2.4 Models

The experiments described in the following chapters rely on several models based on different architectures. In order to evaluate the contribution of attention, we consider both Transformers and architectures not based on attention. In order to assess the contribution of multimodal learning, we take both blind architectures receiving only the textual input and multimodal ones into account. Since the architectures not based on attention are not the main focus of this thesis and they are rather simple, consisting of a combination of LSTMs and CNNs, we describe the way they are adapted to the tasks at hand directly in the following chapters. In the following sections we generally describe

RoBERTa, a blind Transformer, and LXMERT, a multimodal one. The way they are adapted to the tasks at hand is described in the following chapters.

2.4.1 RoBERTa

The Robustly-Optimized version of BERT (Devlin et al., 2019), RoBERTa, is a state-of-the-art Transformer introduced by Liu et al. (2019b). RoBERTa is a classical Transformer model whose training has been performed selecting the parameters which led to the best performance. RoBERTa_{BASE}, the released model, has been pre-trained on 16GB of English text trained for 500K steps in order to perform masked language modeling. Unlike BERT, RoBERTa is not trained on the next sentence prediction task, since the authors shown that their Transformer performed better when pre-trained using only the masked language modeling task. It has 12 self-attention layers with 12 heads each. It uses three special tokens, namely CLS, which is placed at the beginning of the sentence and which is taken to be the representation of the given sequence, SEP, which separates sequences, and EOS, which denotes the end of the input. The representation corresponding to the CLS token can be used as a representation of the whole sentence. Hence, during the fine-tuning of RoBERTa, the CLS representation can be given as input to an MLP in order to generate an answer for the task at hand.

2.4.2 LXMERT

LXMERT (Learning Cross-Modality Encoder Representations from Transformers) (Tan and Bansal, 2019) is a pre-trained multimodal Transformer. It represents an image by the set of position-aware object embeddings for the 36 most salient regions detected by a Faster R-CNN and it processes the text input by position-aware randomly-initialized word embeddings. Both the visual and linguistic representations are processed by a specialized transformer encoder based on self-attention layers; their outputs are then processed by a cross-modality encoder that through a cross-attention mechanism generates representations of the single modality (language and visual output) enhanced with the other modality as well as their joint representation (cross-modality output). LXMERT

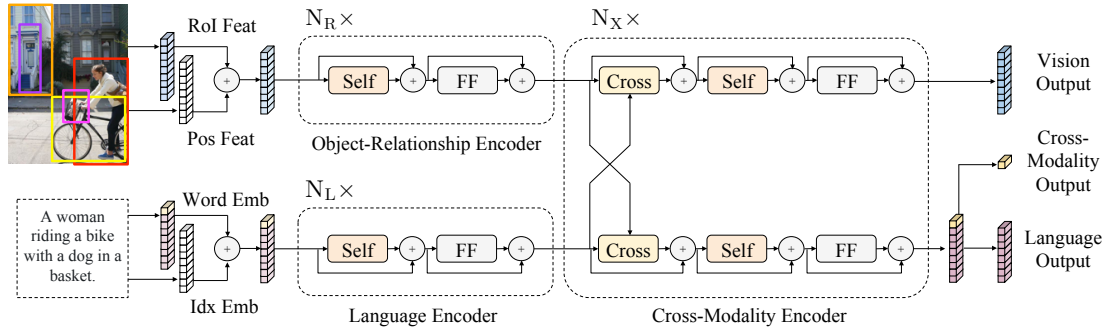


Figure 2.8: Representation of the LXMERT architecture. Figure taken from the paper by [Tan and Bansal \(2019\)](#).

has 19 attention layers: 9 and 5 self-attention layers in the language and visual encoders, respectively, and 5 cross-attention layers. A representation of the LXMERT architecture is shown in figure 2.8. As RoBERTa, LXMERT uses the special tokens CLS and SEP. Differently from RoBERTa, LXMERT uses the special token SEP both to separate sequences and to denote the end of the textual input. LXMERT has been pre-trained on five tasks: masked cross-modality language modeling, masked object prediction via RoI-feature regression, masked object prediction via detected-label classification, cross-modality matching, and image question answering. Cross-modality language modeling requires to predict masked words in a sentence. RoI-feature regression requires to generate the features corresponding to the visual content of some masked objects in the image. Detected-label classification requires to predict the labels corresponding to the category of the masked objects in the image. Cross-modality matching requires to establish whether a sentence is the caption of the given image or not. Finally, Image Question answering requires to predict the answer of the questions about an image. As in RoBERTa, the representation corresponding to the CLS token can be used as a representation of the whole sentence. Hence, it can be given as input to an MLP in order to generate an answer for the task at hand during fine-tuning.

2.5 Tasks and Datasets

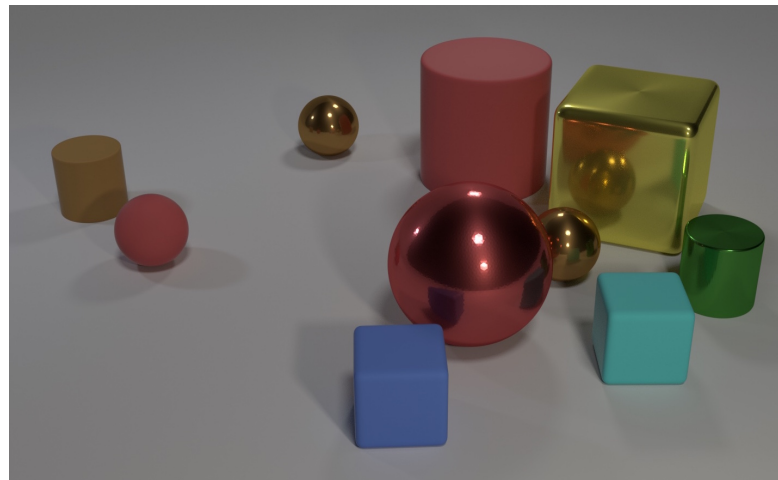
Many tasks have been proposed in order to assess whether models are able to effectively integrate information coming from language and vision. The Image Captioning (IC)

task requires to describe the content of an image with a sentence in natural language (Bernardi et al., 2016a). The task can be formulated both as a retrieval or a generation task. In the former case, given an image and a set of descriptions in natural language, the model must select the sentence which best describes the image (Hodosh et al., 2013; Socher et al., 2014; Farhadi et al., 2010; Ordonez et al., 2011; Jia et al., 2011), whereas in the latter case, given an image, the model must generate a sentence in natural language which properly describes the content of the image (Kulkarni et al., 2013; Vinyals et al., 2015; Karpathy and Fei-Fei, 2015). IC is not easy to evaluate, since many descriptions can properly describe the content of an image. Hence, research in multimodal learning put a lot of effort in proposing multimodal tasks which are easier to be evaluated. The Visual Question Answering (VQA) task consists in providing an answer to a question in natural language about an image (Sharma and Jalal, 2021). The task can be cast both as a multiple choice task where the system has to select the answer from a list of possible ones or as a task where the system has to generate an appropriate answer. The Visual Dialog task requires a model to hold a dialogue with humans in natural language about visual content (Anderson et al., 1991). Recently, several Visual Dialogue tasks have been proposed as referential guessing games in which an agent asks questions about an image to another agent, and the referent they have been speaking about has to be guessed at the end of the game (de Vries et al., 2017b; Das et al., 2017d; He et al., 2017; Haber et al., 2019; Ilinykh et al., 2019b; Udagawa and Aizawa, 2019). The next sections will describe CLEVR, a VQA dataset, and GuessWhat?!, a visual referential game, which are the main multimodal tasks employed in the following chapters.

2.5.1 CLEVR

CLEVR (Johnson et al., 2017a) allows to study the ability of VQA agents. It requires compositional language and basic spatial reasoning skills. Every question in CLEVR is derived by a Functional Program (FP) from a scene graph of the associated image. The scene graph defines the objects and attributes in the image. The FP contains functions corresponding to skills, e.g., querying object attributes or comparing values. Questions are categorized by their type. CLEVR consists of five question types whose answer la-

bels range over 15 attributes, 10 numbers, and “yes”/“no” (in total 27 labels). Figure 2.9 shows an example of an image and some questions from CLEVR. Questions evaluate different aspects of visual reasoning such as attribute identification, counting, comparison, multiple attention, and logical operations. Figure 2.9 shows some statistics from CLEVR. In particular, it shows the number of training, validation, and test examples (top), the comparison of question lengths for different VQA datasets (bottom left), and the distribution of question types in CLEVR (bottom right).



- Q:** Are there an equal number of large things and metal spheres?
- Q:** What size is the brown cylinder that is left of the brown metal thing that is left of the big sphere?
- Q:** There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
- Q:** How many objects are either small cylinders or metal things?

Figure 2.9: Example of image and questions from CLEVR. Questions evaluate different visual reasoning skills. Figure taken from the paper by [Johnson et al. \(2017a\)](#).

2.5.2 GuessWhat?!

The GuessWhat?! dataset was collected via Amazon Mechanical Turk by [de Vries et al. \(2017b\)](#). It is an asymmetric game involving 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 Guesser/Questioner) has to guess it by asking Yes/No questions to the Oracle. There are no time constraints to play the game. The game is considered successful if the target

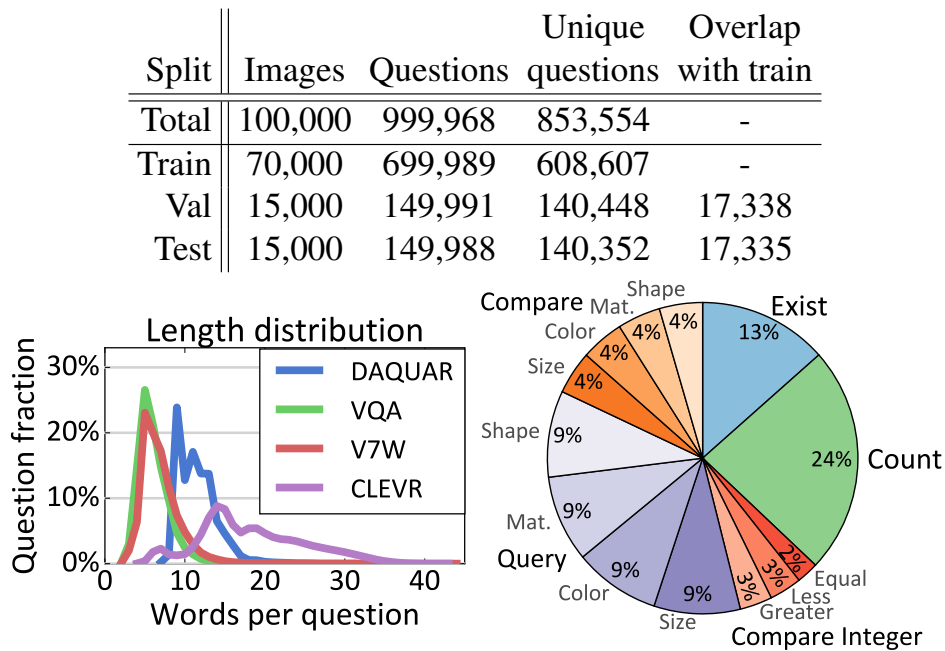


Figure 2.10: **Top:** Statistics for CLEVR. Most questions are unique and few questions from the validation and test sets appear in the training set. **Bottom left:** Comparison of question lengths for different VQA datasets. **Bottom right:** Distribution of question types in CLEVR. Figure and analyses taken from the paper by [Johnson et al. \(2017a\)](#).

object selected by the Oracle has been correctly identified by the Guesser/Questioner.

Figure 2.11 shows an example of a dialogue from GuessWhat?!

Questioner	Oracle
1. Is it on a wooden surface?	Yes
2. Is it red?	No
3. Is it white?	No
4. Is it a scissor?	Yes
5. Is it the scissor on the left of the picture?	Yes

Figure 2.11: GuessWhat?! human dialogues are short and with a clear division of roles between players; most of the last questions are answered positively, are long, and contain details suitable to guess the target object.

The dataset contains 155K English dialogues about approximately 66K different images. The answers are respectively 52.2% No, 45.6% Yes, and 2.2% N/A (not applicable); the training set contains 108K datapoints and the validation and test sets 23K each.

Dialogues contain on average 5.1 (± 3.3) question-answer (QA) pairs and the vocabulary consists of around 4900 words; each game has at least 3 and at most 20 candidates. We evaluate models using human dialogues, selecting only the games on which humans have succeed finding the target and contain at most 10 turns (total number of dialogues used: 90K in training and around 18K both in validation and testing).

In Chapter 3 (Greco et al., 2020) we run a careful analysis of the dataset aiming to find features useful to better understand the performance of models. Although the overall number of Yes/No answers in the dialogues is balanced, the shorter the dialogues, the higher the percentage of Yes answers is: it goes from the 75% in dialogues with 2 turns to the 50% in the 5 turn cluster to the 35% in the 10 turn cluster. Interestingly, most questions in the last turns obtain a positive answer and these questions are on average longer than earlier ones (see Figure 2.11 for an example). A model encoding these questions well has almost all the information to guess the target object without actually using the full dialogue history. Not all games are equally difficult: in shorter dialogues the area of the target object is bigger than the one of target objects in longer dialogues, and their target object is quite often a “person” – the most common target in the dataset; moreover, the number of distractors in longer dialogues is much higher. Hence, the length of a dialogue is a good proxy of the difficulty of the game. Figure 2.12 reports the statistics of the training set; similar ones characterize the validation and the test sets.

The dialogue length is a good proxy of the level of difficulty of the game. Figure 2.13 shows that longer dialogues contain more distractors and in particular more distractors of the same category of the target object; the latter are supposed to be especially challenging for the models, because the usual architecture of the Guesser receives the category and coordinates of each candidate object. Moreover, the area occupied by target objects is smaller in longer dialogues and the most representative category among target objects (“person”) is less frequent. Finally, longer dialogues contain more words occurring rarely in the training set (i.e., words appearing less than 15 times in the training set). We will exploit these features in order to scrutinize the behaviour of models.

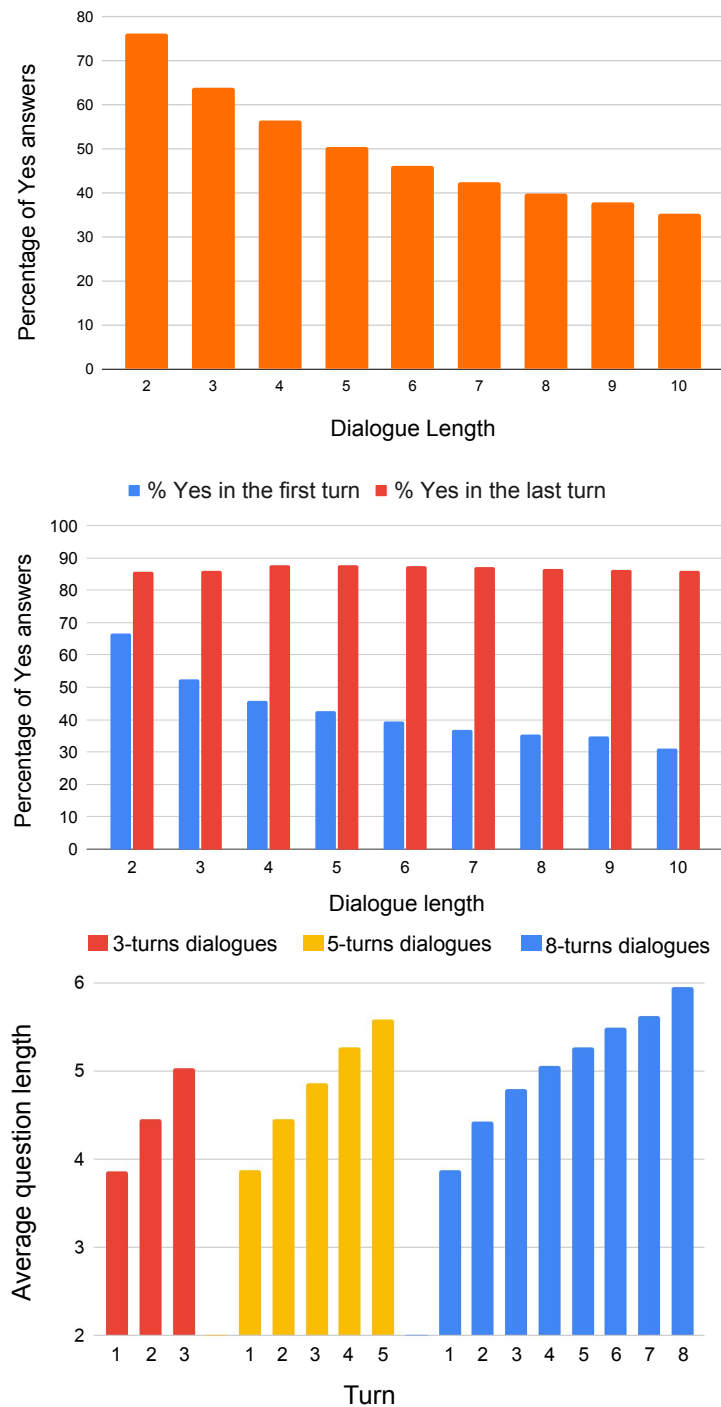


Figure 2.12: Statistics of the training set (the validation and test sets have similar distributions). Dialogue length refers to the number of turns. **Up:** The distribution of Yes/No questions is very unbalanced across the clusters of games (the percentage of Yes answers is much higher in shorter dialogues); **Middle** In the large majority of games, the last question is answered positively; **Bottom:** The last questions are always longer (length of questions per turn for the clusters with dialogues having 3, 5, and 8 turns).

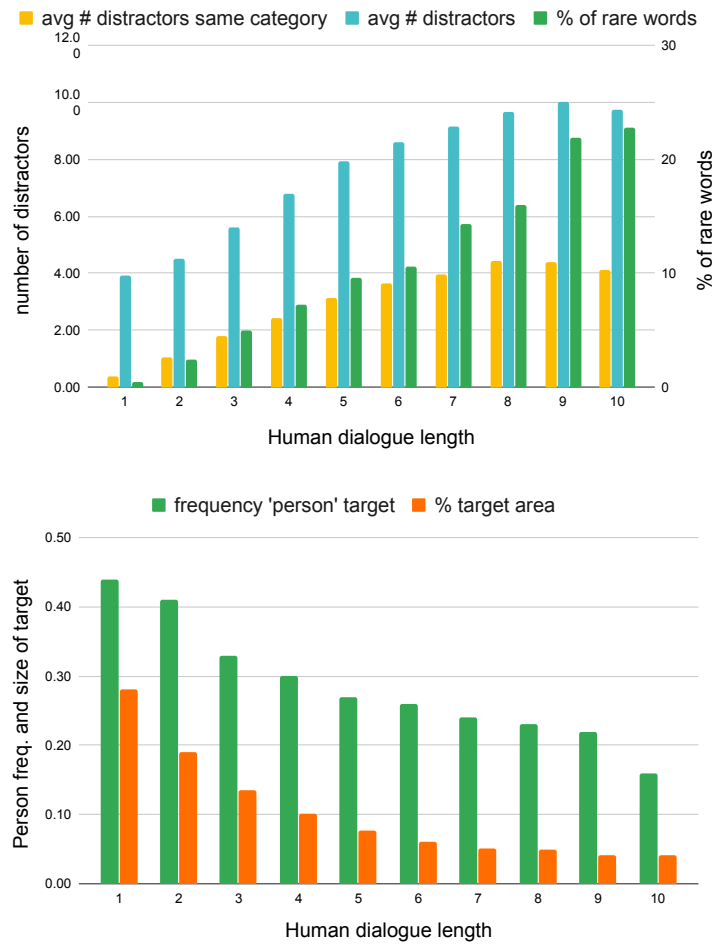


Figure 2.13: **Up**: longer human dialogues contain more distractors and more distractors of the same category of the target object, and more rare words; **Down**: The distribution of target objects is unbalanced, since “person” is the most frequent target.

Chapter 3

Pre-Trained Transformers Encoding the History of a Visual Dialogue

In this chapter, we study the issue of visually grounded dialogue history encoding. We compare models across several dimensions: the architecture (LSTMs vs. Transformers), the input modalities (only language vs. language and vision), and the model background knowledge (trained from scratch vs. pre-trained and then fine-tuned on the downstream task). We show that pre-trained Transformers, RoBERTa and LXMERT, are able to identify the most salient information independently of the order in which the dialogue history is processed. Moreover, we find that RoBERTa handles the dialogue structure to some extent; instead LXMERT can effectively ground short dialogues, but it fails in processing longer dialogues having a more complex structure.

3.1 Introduction

Visual Dialogue tasks have a long tradition (e.g. [Anderson et al., 1991](#)). Recently, several dialogue tasks have been proposed as referential guessing games in which an agent

This chapter describes the work by [Greco et al. \(2020\)](#).

asks questions about an image to another agent and the referent they have been speaking about has to be guessed at the end of the game (de Vries et al., 2017b; Das et al., 2017d; He et al., 2017; Haber et al., 2019; Ilinykh et al., 2019b; Udagawa and Aizawa, 2019). Among these games, GuessWhat?! and GuessWhich (de Vries et al., 2017b; Das et al., 2017d) are asymmetrical – the roles are fixed: one player asks questions (the Questioner) and the other (the Oracle) answers. The game is considered successful if the Guesser, which can be the Questioner itself or a third player, selects the correct target.

Most Visual Dialogue systems proposed in the literature share the encoder-decoder architecture (Sutskever et al., 2014b) and are evaluated using the task-success of the Guesser. By using this metric, multiple components are evaluated at once: the ability of the Questioner to ask informative questions, of the Oracle to answer them, of the Encoder to produce a visually grounded representation of the dialogue history and of the Guesser to select the most probable target object given the image and dialogue history.

We disentangle the compressed task-success evaluation and focus on the ability of the Encoder to produce a dialogue hidden state representation that encodes the information necessary for the Guesser to select the target object. Hence, we use the dialogue history generated by humans playing the referential game so to be sure of the quality of the questions and of the answers. We run our analysis on GuessWhat?! since its dialogues are quite simple: a sequence of short questions answered by Yes or No containing on average $30.1 (\pm 17.6)$ tokens per dialogue. The simplicity of the dialogue structure makes the dataset suitable to be used as a diagnostic dataset.

In Sankar et al. (2019), the authors have shown that neural models are not sensitive to the order of turns in dialogues and conclude they do not use the history effectively. In GuessWhat?! dialogues the order in which questions have been asked is not crucial: we would be able to guess the target object even if the question-answer pairs in Figure 2.11 were provided in the reversed order. Indeed, we are able to use salient information independently of the turns where it occurs. We wonder whether the same holds for neural models trained to solve the GuessWhat?! task. As the example in the figure shows, the last question humans ask is quite rich in detail about the target object and is answered positively. We exploit these features of the dataset to run our in-depth analysis.

We compare encoders with respect to the architecture (Recurrent Neural Networks vs. Transformers), the input modalities (only language vs. language and vision), and the model background knowledge (trained from scratch vs. pre-trained and then fine-tuned on the downstream task). Our analysis shows that:

- Transformers are less sensitive than Recurrent Neural Network based models to the order in which QA pairs are provided;
- pre-trained Transformers detect salient information, within the dialogue history, independently of the position in which it is provided;
- LXMERT outperforms RoBERTa on shorter dialogues, but it struggles in processing longer ones where the dialogue structure plays a major role.

3.2 Related Work

Scrutinizing Visual Dialogues Encoding Interesting exploratory analysis has been carried out in order to understand VQA systems and highlight their weaknesses and strengths, (e.g. [Johnson et al., 2017c](#); [Shekhar et al., 2017a](#); [Suhr et al., 2017](#); [Kafle and Kanan, 2017](#)). However, less is known about how well grounded neural conversational models are able to effectively encode the received dialogue history.

In [Sankar et al. \(2019\)](#), the authors study how neural dialogue models encode the dialogue history when generating the next utterance. They show that neither recurrent nor transformer based architectures are sensitive to perturbations in the dialogue history and that Transformers are less sensitive than recurrent models to perturbations that scramble the conversational structure; furthermore, their findings suggest that models enhanced with attention mechanisms use more information from the dialogue history than their vanilla counterpart. We take inspiration from this study in order to understand how well state-of-the-art models encode the visually grounded dialogues generated by humans while playing the role of the Guesser in the GuessWhat?! game.

In [Kaushik and Lipton \(2018\)](#), the authors show that in many reading comprehension datasets, that presumably require the combination of both questions and passages to predict the correct answer, models can achieve quite a good accuracy by using only part of the information provided. We investigate the role of each turn in GuessWhat?! human dialogues and to what extent models encode the strategy seen during training.

SOTA LSTM Based Models on GuessWhat?! After the introduction of the supervised baseline model ([de Vries et al., 2017b](#)), several models have been proposed to play the GuessWhat?! game. They exploit either some form of reinforcement learning ([Sang-Woo et al., 2019](#); [Zhang et al., 2018b](#); [Zhao and Tresp, 2018](#); [Zhang et al., 2018a](#); [Gan et al., 2019](#); [Yang et al., 2019](#); [Pang and Wang, 2020](#)) or cooperative learning ([Shekhar et al., 2019](#); [Pang and Wang, 2020](#)); in both cases, the model is first trained with the supervised learning regime and then the new paradigm is applied. This two-step process has been shown to reach higher task success than the supervised approach when the Questioner and Oracle models are put to play together. Since our focus is on the Guesser and we are evaluating it on human dialogues, we will compare models that have undergone only the supervised training step. We compare these recurrent models (based on LSTMs) against models based on Transformers ([Vaswani et al., 2017](#)).

Transformer Based Models [Vaswani et al. \(2017\)](#) showed the power of the attention mechanisms at the core of Transformers. The last years have seen an increasing popularity of these models trained on several tasks to reach task-agnostic multimodal representations ([Li et al., 2019](#); [Lu et al., 2019a](#); [Tan and Bansal, 2019](#); [Chen et al., 2019a](#); [Su et al., 2020](#); and [Nan Duan et al., 2020](#)). ViLBERT ([Lu et al., 2019a](#)) has been recently extended by means of multi-task training involving 12 datasets which include GuessWhat?! ([Lu et al., 2020](#)) and has been fine-tuned to play the Answerer of VisDial ([Murahari et al., 2019a](#)). Among these universal multimodal models, we choose LXMERT ([Tan and Bansal, 2019](#)). [Clark et al. \(2019\)](#) propose methods for directly analyzing the attention heads aiming to understand whether they specialize in some specific foundational aspect (like syntactic relations) functional to the overall success of the model. We take inspiration from their work in order to shed light on how

Transformers, that we adapt to play GuessWhat?!, encode the dialogues.

3.3 Dataset

We rely on GuessWhat?!, a referential game involving a dialogue about images previously described in Section 2.5.2. In particular, given the human dialogue history, the image, and a list of candidate objects, we are interested in assessing whether multimodal models are able to select the object the dialogue is talking about.

3.4 Models

All the evaluated models share the skeleton as illustrated in Figure 3.1: an encoder paired with a Guesser. For the latter, all models use the module proposed in [de Vries et al. \(2017b\)](#). Candidate objects are represented by the embeddings obtained via a MLP starting from the category and spatial coordinates of each candidate object. The obtained representations are used to compute dot products with the hidden dialogue state produced by an encoder. The scores of each candidate object are given to a softmax classifier in order to choose the object with the highest probability. The Guesser is trained in a supervised learning paradigm, receiving the complete human dialogue history at once. The models we compare differ in how the hidden dialogue state is computed. We compare LSTMs vs. Transformers when receiving only the language input (henceforth, Blind models) or both the language and the visual input (henceforth, Multimodal models).

3.4.1 Language-only Encoders

LSTM As in [de Vries et al. \(2017b\)](#), the representations of the candidates are fused with the last hidden state obtained by an LSTM which processes only the dialogue history.

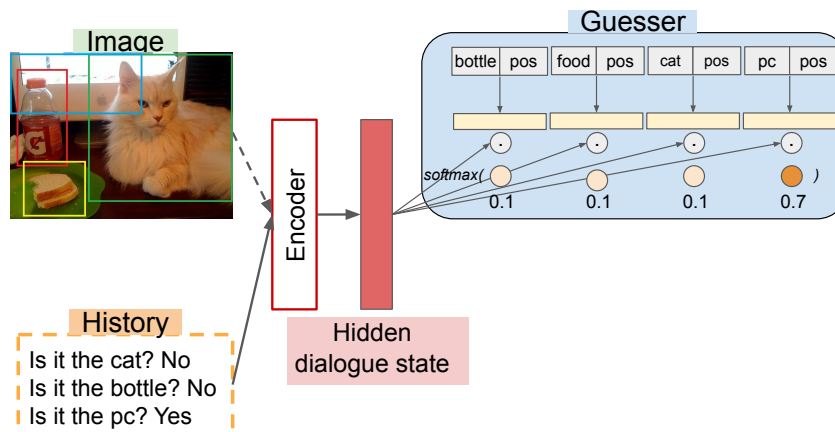


Figure 3.1: Shared Encoder-Guesser skeleton. The Guesser receives the category labels (e.g., “bottle”) and the spatial coordinates (pos) of each candidate object. Multimodal encoders receive both the image and the dialogue history, whereas blind models receive only the latter.

RoBERTa In the architecture of the model described above, we replace the LSTM with RoBERTa (Liu et al., 2019b).¹ We use RoBERTa_{BASE} which has been pre-trained on 16GB of English text for 500K steps in order to perform the masked language modeling task. We give the output corresponding to the CLS token to a linear layer and a *tanh* activation function to obtain the hidden state which is given to the Guesser. To study the impact of the pre-training phase, we have compared the publicly available pre-trained model, which we fine-tuned on GuessWhat?! (**RoBERTa**), against its counterpart trained from scratch only on the game (**RoBERTa-S**).

3.4.2 Multimodal Encoders

V-LSTM We enhance the LSTM model described above with the visual modality by concatenating the linguistic and visual representation and scaling its result with an MLP; the result is passed through a linear layer and a *tanh* activation function to obtain the hidden state which is used as input for the Guesser modules. We use a frozen ResNet-152 pre-trained on ImageNet (He et al., 2016) to extract the visual vectors.

¹We have also tried BERT, but we obtained a higher accuracy with RoBERTa.

LXMERT In order to evaluate the performance of a multimodal pre-trained Transformer, we employ LXMERT (Tan and Bansal, 2019). We process the representation corresponding to the CLS token as in RoBERTa. Similarly, we consider both the pre-trained version (**LXMERT**) and its counterpart trained from scratch (**LXMERT-S**).

3.5 Experiments

We compare the models described above using human dialogues aiming to shed light on how the encoders capture the information that is salient to guess the target object.

3.5.1 Task Success

As we can see in Table 3.1, the pre-trained Transformers LXMERT and RoBERTa obtain the highest results, with the multimodal model scoring slightly higher (69.2 vs. 67.9).² The high accuracy obtained by RoBERTa shows that the dialogue history per se is quite informative to select the right target object. If we go back to the example in Figure 2.11, we realize it is possible to succeed in that game if we are given the dialogue only and are asked to select the target object (the scissor on the left) among candidates for which we are told the category and the coordinates – as it is the case for the Guesser.

The comparison between the pre-trained version of these models with their from-scratch counterparts highlights the role of the pre-training in language understanding (RoBERTa vs. RoBERTa-S) and in language grounding (LXMERT vs. LXMERT-S). To better understand the difference between the models, Table 3.1 reports also the accuracy by clusters of games based on the dialogue length. Quite interestingly LXMERT performs very well on short dialogues: it reaches 80.5% accuracy on 3-turn dialogues, but it has a rather big drop when dialogues get longer. The difference between LXMERT and LXMERT-S is minimal for the 8-turn cluster. Instead, RoBERTa is less affected by the length of the dialogues. This difference between the two pre-trained transformers

²The model proposed in Lu et al. (2020) based on ViLBERT obtains an accuracy on GuessWhat?! with human dialogues of 65.04% when trained together with the other 11 tasks and 62.81% when trained only on it.

	LSTM	RoBERTa-S	RoBERTa	V-LSTM	LXMERT-S	LXMERT
All	64.7	64.2	67.9	64.5	64.4	69.2
3	72.5	72.7	75.3	71.9	72.7	80.5
5	59.3	58.3	60.1	59.3	58.9	63.1
8	47.3	45.1	51.0	47.2	46.1	45.0

Table 3.1: Model comparison on the accuracy results for all games, and for those of 3/5/8 dialogue length.

suggests that LXMERT is good in exploiting language grounding when the dialogue (and maybe also the image) is not too complex, while RoBERTa can handle the dialogue structure to some extent.

In the following, we are running an in-depth analysis to understand whether models are able to identify salient information independently of the position in which they occur.

3.5.2 Are Models Sensitive to the Strategy Seen during Training?

In Section 3.3, we have seen that human dialogues tend to share a specific strategy, i.e. questions that are asked in first turns are rather short whereas those in the last turns provide relevant details about the most probable target object. We wonder whether the models under analysis become sensitive to the above-mentioned strategy and learn to focus on some turns more than others rather than on the actual salient QA pair.

Inspired by Sankar et al. (2019), we perturb the dialogue history in the test set by reversing the order of turns from the last to the first one. Differently from them, given the nature of the GuessWhat?! dialogue history, we value positively models that are robust to this change in the dialogue history order. In the following, we refer to the dialogues provided in the order asked by humans as Ground Truth (GT) and to the dialogues provided in the reverse order as Reversed.

Our experiment (Table 3.2) shows that Transformers are less sensitive than LSTMs to the order in which QA pairs are provided. Interestingly, the pre-training phase seems to mitigate the effect of the change of the order even more. Indeed, RoBERTa has a drop of just 1.4, whereas the accuracy of its from-scratch counterpart drops of 6.4. The difference is even more noticeable in the case of LXMERT: while LXMERT has a drop

		GT	Reversed
BLIND	LSTM	64.7	56.0
	RoBERTa-S	64.2	57.8
	RoBERTa	67.9	66.5
MM	V-LSTM	64.5	51.3
	LXMERT-S	64.4	57.8
	LXMERT	69.2	65.1

Table 3.2: Accuracy obtained on the test set containing dialogues in the Ground Truth order (GT) vs. the reversed order (Reversed).

of 4.1, the accuracy of its from-scratch counterpart drops of 6.6%. In other words, (pre-trained) Transformers seem to be able to identify salient information independently of the position in which it is provided within the dialogue history.

3.5.3 The Role of the Last Question

Table 3.3 reports the results of the models when receiving all the turns of the dialogue history, when receiving the dialogue history without the last turn, and when receiving only the last turn. As we can see all models undergo a rather big drop in accuracy when removing the last question. It is worth noting that RoBERTa outperforms other models when removing the last turn, confirming that RoBERTa is able to better encode the full dialogue history and not only parts of it. This holds for different dialogue lengths as shown in the Table. Interestingly, LXMERT performs quite well in short dialogues also when given only the last question: it reaches already 68.6% in the 3-turn cluster, namely +7.6 than RoBERTa. Instead, with longer dialogues it does not manage to exploit the last question so well reaching an accuracy closer to RoBERTa’s (32.3 vs. 30.1). By comparing the accuracy of each model when receiving only the last turn and when receiving all turns except the last one, we can notice an interesting pattern: whereas in short dialogues models obtain a rather high accuracy when receiving either only the last question or only the previous turns, they are able to profit of the last turn much less in longer dialogues. This could be due to the fact that in short dialogues the last question describes the target object without relying on too many information stated far away on previous turns.

Model	3-Q			5-Q			8-Q		
	All	W/o last	Last	All	W/o last	Last	All	W/o last	Last
LSTM	72.5	53.4	56.9	59.3	46.8	39.3	47.3	38.4	26.7
RoBERTa-S	72.7	55.4	55.3	58.3	44.9	37.4	45.0	38.9	27.6
RoBERTa	75.3	58.2	61.0	60.1	49.3	39.4	51.0	42.0	30.1
V-LSTM	71.9	53.8	53.0	59.3	43.7	34.0	47.2	36.5	21.9
LXMERT-S	72.7	55.4	56.7	58.9	46.9	38.7	46.1	39.7	28.8
LXMERT	80.5	56.8	68.6	63.1	47.7	46.0	45.0	37.7	32.3

Table 3.3: Accuracy of the models when receiving all turns of the dialogue history and when removing the last turn (W/o last) or receiving only the last turn (Last) for dialogues with 3, 5, and 8 turns.

3.5.4 How Attention is Distributed across Turns

So far we have seen that the last turn is usually answered positively (Section 3.3) and that it is quite informative to detect the target object (Section 3.5.1). We wonder whether this is reflected on how models distribute their attention across turns within a dialogue. To this end, we analyze how much each turn contributes to the overall self-attention within a dialogue by summing the attention of each token within a turn. We run this analysis for LXMERT and RoBERTa in their various versions: all models put more attention on the last turn when the GT order of turns is given.

In Table 3.2, we have seen that Transformers are more robust than the other models when the dialogue history is presented in the reversed order (the first QA pair of the GT is presented as the last turn and the last QA pair is presented as first turn). Our analysis of the attention heads of RoBERTa and LXMERT shows that these models, both in their from scratch and pre-trained version, focus more on the question *asked* last **also in the reversed test set** where it is *presented* in the first position. This shows they are still able to identify the most salient information. In Figure 3.2, we report the attention per turn of LXMERT-S when receiving the GT and the reversed test set in 5-turn dialogues.

3.5.5 Qualitative Evaluation

The quantitative analysis reported so far shows that the pre-trained transformers, LXMERT and RoBERTa, overall have a similar performance, but that LXMERT is much better in exploiting the last question in short dialogues and fails encoding the information pro-

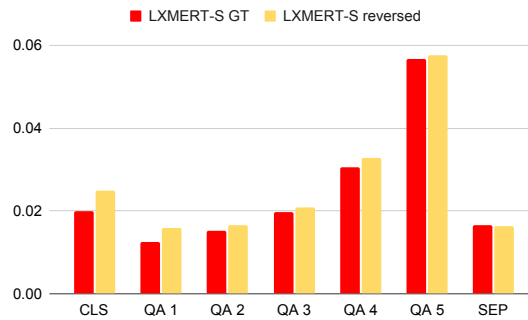
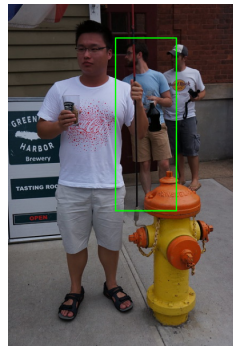


Figure 3.2: Attention assigned by LXMERT-S to each turn in a dialogue when the dialogue history is given in the GT order (from QA1 to QA5) or in the reversed order (from QA5 to QA1).

vided by long dialogues. RoBERTa instead is affected less by the dialogue length and takes less advantage of the informative question asked in the last turn by humans. In order to gain a deeper understanding about the differences between these two models, we analyzed games which are solved successfully by RoBERTa and not by LXMERT and vice-versa. Dialogues solved by RoBERTa and not by LXMERT have a mean length of $5.5 (\pm 2.3)$, whereas dialogues belonging to the opposite case have a mean length of $4.5 (\pm 2.0)$. This confirms the hypothesis that RoBERTa encodes longer dialogues better than LXMERT. The qualitative analysis shows that LXMERT has an advantage when dealing with shorter dialogues that require to rely on vision.

In Figure 3.3, we show two examples of dialogues one which has been solved by LXMERT and not by RoBERTa (left) and one solved by RoBERTa but not by LXMERT (right). In the dialogue on the left, the model needs to ground the question “Is he wearing blue?” in the image to properly process it. LXMERT succeeds in this game. This suggests that though the Guesser does not see the candidate visual representation it manages to profit of the language grounding ability of the encoder. In the dialogue on the right, the model needs to properly solve the anaphora in the last question “Is it in the back?” connecting the pronoun to the “car” mentioned in the second turn. LXMERT fails establishing such connection whereas RoBERTa seems to succeed in solving the anaphora.



Questioner	Oracle
1. Is it a person?	Yes
2. Is he in the foreground?	No
3. Is he wearing blue?	Yes



Questioner	Oracle
1. Is it a sign?	No
2. Is it a car?	Yes
3. Is it white?	No
4. Is it in the middle?	No
5. Is it in the back?	Yes

Figure 3.3: A game solved successfully by LXMERT and not by RoBERTa (left) and a game solved by RoBERTa and not by LXMERT (right).

3.6 Conclusion

Our comparative analysis has shown that Transformers are less sensitive than LSTMs to the order in which QA pairs are provided and that their pre-trained versions are even stronger in detecting salient information, within the dialogue history, independently of the position in which it is provided.

We also shown that RoBERTa is the encoder providing the Guesser with the most informative representation of the dialogue history. Its advantage is particularly strong in longer dialogues. On the other hand, LXMERT greatly outperforms all the other models on 3-turn dialogues: indeed, it succeeds in providing the Guesser with a grounded representation of the dialogue history when the latter consists of a few turns while it fails in doing so for longer dialogues. All our models currently rely on categories to represent candidate objects in the Guesser. It would be interesting to see how models would perform when they have to rely on visual information rather than categories.

3.7 Summary

So far, we discovered that:

- + Pre-Trained Transformers understand polar question-answer pairs;
- + Pre-Trained Transformers pinpoint the most important dialogue turns;
- + Pre-training (more than the attention mechanism) is the feature which makes the biggest difference in accuracy with respect to the other models.

Chapter 4

Pre-trained Transformers Grounding Different Spatial Questions

In this chapter, we study the grounding skills required in order to answer spatial questions about a target object in an image. We propose a classification for spatial questions dividing them into absolute, relational, and group questions. We build an answerer model based on a pre-trained multimodal Transformer and we compare it with its counterpart trained from scratch and a baseline with and without visual features of the scene. We are interested in studying how the attention mechanism of the pre-trained multimodal Transformer is used to answer spatial questions since they require paying attention on more than one region simultaneously and spotting the relation holding among them. We show that the model based on a pre-trained multimodal Transformer outperforms the baselines by a large extent. By analyzing the errors and the attention mechanism of the multimodal pre-trained Transformer, we find that our classification helps to gain a better understanding of the skills required to answer different spatial questions.

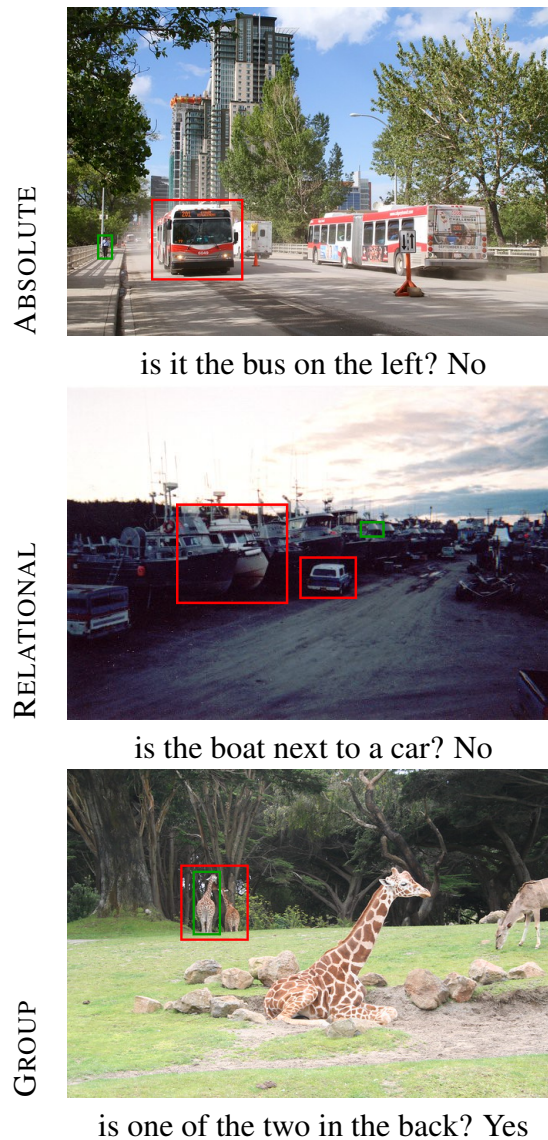


Figure 4.1: A vast amount of questions asked by humans in the GuessWhat?! game (de Vries et al., 2017a) are spatial. We classify them as *absolute*, *relational*, and *group* based on how many objects are involved and how they are related. The red box marks the object(s) involved in the question, while the green box marks the target of the game. *Relational* and *group* questions need more than one object, whereas *absolute* do not.

4.1 Introduction

Visual Dialogues are a useful testbed to study how models ground natural language and in particular how they ground spatial language, which is the focus of our analysis.

This chapter describes the work by Testoni et al. (2020).

Visual Dialogues have been the aim of early work on natural language understanding (NLU) (Winograd, 1972) and are now studied by a very active community at the interplay between computer vision and computational linguistics (e.g. Baldrige et al., 2018; Ilinykh et al., 2019b; Haber et al., 2019). Recently, important progress has been made on visual dialogue systems thanks to the release of datasets like VisDial (Das et al., 2017a) and GuessWhat?! (de Vries et al., 2017a). The former contains chit-chat conversations about an image whereas the latter is a visual game, hence its dialogues are goal-oriented. In both cases, one agent asks questions and the other, which we call the Oracle, answers. For VisDial most of the work focused on the answerer, but in-depth evaluation has been carried out on the questioner too (e.g. Murahari et al., 2019b; Testoni et al., 2019). For GuessWhat?!, instead, work has been done mostly, if not only, on the questioner. Current models trained with reinforcement learning achieve high task success; they adapt to the oracle limitations and end-up asking questions that are linguistically simpler than those asked by humans (Shekhar et al., 2019; Pang and Wang, 2020).

It is interesting to understand where current multimodal NLU models stand with respect to this task: answering questions asked by humans in a goal oriented visual dialogue. This chapter addresses this question by evaluating how the Oracle model of the Guess-What?! game answers questions asked by humans while playing the game.

We rely on the GuessWhat?! visual referential game described in Section 2.5.2 and we focus on the role of the Oracle, which answers questions about a target object in the image. Shekhar et al. (2019) show that most of the questions in the dataset are about the entity of the target (“Is it a female?”) or its location (“is it the first one?”). Mazuecos et al. (2020) show that the baseline model, commonly used for the Guesswhat?! task since its introduction in de Vries et al. (2017a), has almost human-like accuracy on the entity questions and a much lower accuracy on questions about attributes. In this chapter, we focus on spatial questions and classify them into three groups: *absolute*, *relational*, and *group* questions as illustrated in Figure 4.1.

An unpleasant aspect of the baseline model is that it receives the gold standard entity of the target (that is, the category label, e.g. “giraffe” or “boat”) as input. Furthermore, it

answers questions without seeing either the image or the visual features of the target, but instead it simply relies on the category label of the target and its coordinates. Important progress on multimodal encoders has been obtained since the GuessWhat?! release; hence, we study the effect of using models that ground the question into the image and do not have access to the gold standard category label of the target. We adapt a multimodal universal encoder, LXMERT (Tan and Bansal, 2019), to play the role of the Oracle and compare it with the baseline model.

It is known that grounding spatial expressions is challenging for neural networks since quite often they require models to put attention on more regions simultaneously and spot the relation holding among them (e.g., the car and the boat in Figure 4.1, middle). LXMERT is a transformer-based neural network and as such it heavily exploits attention-based mechanisms. In this chapter, we run a qualitative analysis of the attention LXMERT exhibits for the different types of location questions and run an in-depth error analysis of its results. To sum up, we make the following contributions:

- We adapt LXMERT to play the role of the Oracle of the GuessWhat?! game obtaining an overall accuracy of 82.21%, an increase of 6.27% with respect to the usual baseline;
- We find that LXMERT improves over the baseline also on spatial questions (+9.70%), but they remain a large source of errors also for this model – with 77.00% accuracy;
- We classify spatial questions into three sub-types and use this classification to annotate the subset of spatial questions in the GuessWhat?! test set. The fine-grained evaluation shows that the hardest spatial questions are the *relational* and *group* ones;
- We run an in-depth qualitative analysis of LXMERT cross-modal attention and an analysis of its errors on each question sub-type. The analysis shows that LXMERT attention differs between *absolute* and *relational* questions as expected, and that some spatial questions need the dialogue history to be interpreted correctly.

The chapter is organized as follows. Section 4.2 reviews previous work on visual question answering and on spatial referring expressions. Section 4.3 presents the models providing information on how we adapt LXMERT for the Oracle task. Section 4.4 describes the dataset and our classification of spatial questions. In Section 4.5 we compare the accuracy of the models reporting a fine-grained evaluation by question type and zoom into the subset of spatial questions. We further analyzed this subset through a manual inspection of LXMERT attention and errors in Section 4.6, before drawing our conclusions in Section 4.7.

4.2 Related Work

After the introduction of the supervised baseline models (de Vries et al., 2017a), several models have been proposed for the Questioner, which are mostly based on reinforcement learning (Sang-Woo et al., 2019; Zhang et al., 2018b; Zhao and Tresp, 2018; Zhang et al., 2018a; Gan et al., 2019; Yang et al., 2019; Pang and Wang, 2020). For these models, the role of the Oracle is even more salient than for models based on supervised or cooperative learning (Shekhar et al., 2019) since they are reinforced to ask those questions the Oracle is good at answering. Despite this important role of the Oracle, no work has been carried out to evaluate and improve it. We aim to fill this gap.

Shekhar et al. (2019) show that GuessWhat?! human players ask quite a lot spatial questions. It has been observed that capturing the spatial relation about objects is challenging for neural network models. Kelleher and Dobnik (2017) argue that Convolutional Neural Network (CNN) do not ground spatial information properly: since they discard location information through the pooling mechanism, their embeddings can only capture rough relative positions of objects within a scene. In line with this claim, Collell and Moens (2018) show that linguistic features are more spatially informative than CNN visual features. New multimodal models, like LXMERT, start from positional aware embeddings. We therefore study how well they handle the spatial questions asked by GuessWhat?! players.

Spatial expressions have been deeply studied within the referring expression genera-

tion community. In this area, earlier work (Paraboni et al., 2007) has suggested that, in ordered domains (e.g., a document divided into sections and subsections), referring expressions that include spatial information, even when redundant, lead to a significant reduction in the amount of search that is needed to identify the referent. It has been argued that spatial information reduces the cognitive load (measured by eye tracking) necessary for resolving a referring expression (Paraboni et al., 2017). In this research area Krahmer and van Deemter (2012); Ghanimifard and Dobnik (2017) distinguish between spatial referring expressions that involve another object in the description (e.g. “the rabbit in the hat”) from those that do not (e.g. “the rabbit on the left”). The first group of expressions is known as *relational*, while we shall refer to the second one as *absolute*. A further distinction is made between referring expressions that are singular (e.g. “the rabbit in the hat”) and those that are plural (e.g. “the three rabbits on the table”) and refer to a *group* (Lønning, 1997; Gatt and van Deemter, 2007; Krahmer and van Deemter, 2012).

In this chapter, we classify GuessWhat?! spatial questions using *absolute*, *relational* and *group* distinctions and examine how LXMERT performs for each type of spatial question. We also conduct an error analysis and an attention analysis taking these categories into consideration.

Recent work by Agarwal et al. (2020) shows that in current visual dialogue datasets the dialogue history rarely matters. The authors ask crowdsourcers whether they can confidently answer a question by looking at the image and the question, without seeing the dialogue history. In our qualitative analysis we check whether history plays a role for the spatial questions of the GuessWhat?! game that LXMERT fails to answer.

4.3 Models

In this section we present the models that we compare. We also explain how we adapted LXMERT to the Oracle task. The models are trained on successful games.

LSTM is the baseline model proposed in [de Vries et al. \(2017a\)](#). It does not have access to the raw image features. It receives as input embeddings of the target object’s category, its spatial coordinates, and one question encoded by a dedicated LSTM. These three embeddings are concatenated and fed to a Multi-Layer Perceptron (MLP) that gives an answer (Yes or No).

V-LSTM We enhance the LSTM model described above with the visual modality and we remove the information about the target object category. We extract the visual vectors corresponding to the input image and the crop of the target object using a frozen ResNet-152 network pre-trained on ImageNet ([He et al., 2016](#)) and we pass them through a linear layer and a *tanh* activation function. We concatenate these scaled representations to the embeddings of the target object’s spatial coordinates and the question: the resulting vector is fed to an MLP to obtain the answer, as it happens in the LSTM model.

LXMERT We employ LXMERT as our pre-trained multimodal Transformer ([Tan and Bansal, 2019](#)). We process the output corresponding to the CLS token. We consider both the pre-trained version (**LXMERT**) and the one trained from scratch (**LXMERT-S**).¹

4.4 The Dataset

We rely on GuessWhat?!, a referential game involving a dialogue about images previously described in section 2.5.2. In particular, given a question, an image, and a target object, we are interested in assessing whether multimodal models are able to properly answer the question about the target object in the image, playing the role of the Oracle.

[Shekhar et al. \(2019\)](#) propose a classification of the questions based on their focus distinguishing questions which ask about the entity of the target (“Is it an animal?” or “Is

¹We have also evaluated a simplified version of LXMERT-S in which we use 6 self (4 language and 2 visual) and 2 cross-modal attention layers. The model behaves similarly to the more complex version trained from scratch.

		Single label	Multi labels
	Entity	39269	39269
	Not classified	7925	7925
ATTRIBUTE	Spatial	29845	39250
	Color	7145	15403
	Action	3063	7645
	Size	532	1364
	Texture	538	901
	Shape	166	301

Table 4.1: Question type distribution in successful games following the classification proposed in [Shekhar et al. \(2019\)](#) where a question can be assigned to more than one attribute type (multiple labels); the Single label column reports the number of questions which have been assigned to only one type.

it a dog?") or an attribute of it. A question can focus on just one attribute (e.g., "Is it the black dog"? or "Is it black?") in which case it is assigned just to one attribute question type (color in the examples) or about more attributes (e.g., "does it have orange pillows on it?") in which case it is assigned to more attribute question types (to both color and spatial information in the example.) Table 4.1 reports their distribution in the human-human dialogues giving the numbers of questions assigned to one or more types (multi label) or to just one type (single label).

We conjecture that the spatial question type includes questions posing different challenges to multimodal models. [Krahmer and van Deemter \(2012\)](#) divide spatial expressions into *relational* (e.g. "the rabbit in the hat"), that specifies the location of the referent of a noun phrase (the target, "rabbit") relative to another object (the landmark, "hat"), and *absolute* that focus only on the target by providing locative information about it (e.g. "the rabbit on the left"). A third spatial expression that has received attention within the REG community are group referring expressions whose target is a group of entities (e.g. "the three rabbits on the table") or some specific entity of a the group to which the expression refers by ordering them (e.g. "the second rabbit from the left").

We adapt such classification to the GuessWhat?! spatial questions and classify them into four types: relational, absolute, group and other. To distinguish these types we have leveraged syntactic and lexical characteristics specific to each. Relational questions usually include a prepositional phrase followed by a noun phrase that includes either

	%	Example
Relational	31.9	Is it the pen behind the PC?
Absolute	31.8	Is it the one on the left?
Group	17.3	Is it among the 4 women?
Other	19.0	Can you sleep on it?

Table 4.2: Sub-type spatial questions distribution in successful games of questions annotated with only the spatial label in the test set (total: 29845).

a pronoun (e.g. “Is there a sink directly above it?”) or an object word (e.g. “is it the pen behind the laptop?”). Absolute spatial questions (e.g. “the one on the left?”) instead contain a location word either in the x axis (e.g. right, middle, left), or the y (top, bottom), or the z (e.g. front, back) axis. We also consider absolute those questions that include a spatial adjective in its superlative form (e.g. “the leftmost one?”). Finally, we consider group questions those containing a number which may indicate order (e.g. “right to left, is it the first one?”) or groups (e.g. “in the back among four women?”). We have automatically annotated spatial questions by identifying nouns, prepositions and number using the Part of Speech tagger Stanza [Qi et al. \(2020\)](#). When a question is not assigned to any of the three groups, we include it in the “Other” category.² We tried identifying objects using the entity recognizers included in Stanford core NLP ([Manning et al., 2014](#)) and Stanza ([Qi et al., 2020](#)) but the coverage was not good.

In the next section, we will first compare models using the multi-label classification reported in [Table 4.1](#), then we will zoom into the spatial questions which together with the entity questions constitute the large majority of questions asked by humans. In order to understand strength and limits of multimodal models in answering spatial questions, we focus on those which are assigned only to the spatial question type to avoid confounding effects. [Table 4.2](#) reports number of such sub-set.

²Examples of questions following into the “Other” category are: “Is it the tree outside?” – i.e. an elliptical question which could be completed as “Is it the tree outside the fenced garden?” – or “Can you sleep on it?” which is not about a spatial property that occurs in the image but an afforded one.

	LSTM	V-LSTM	LXMERT-S	LXMERT
Entity	93.37	83.24 (-10.13)	88.64 (-4.73)	91.09 (-2.28)
Spatial	67.30	66.40 (-0.90)	71.31 (+4.01)	77.00 (+9.70)
Color	61.64	68.06 (+6.42)	70.51 (+8.87)	76.42 (+14.78)
Action	64.32	65.44 (+1.12)	70.23 (+5.91)	77.16 (+12.84)
Size	60.41	62.76 (+2.35)	67.23 (+6.82)	75.44 (+15.03)
Texture	69.92	66.15 (-3.77)	71.92 (+2.00)	77.47 (+7.55)
Shape	68.44	64.12 (-4.32)	70.76 (+2.32)	74.42 (+5.98)
Not classified	75.02	70.45 (-4.57)	74.94 (-0.08)	82.18 (+7.16)
Total	75.94	72.70 (-3.24)	77.41 (+1.47)	82.21 (+6.27)

Table 4.3: Accuracy of the models on the successful games by question type based on the multi label assignment. Values in parenthesis report the comparison with LSTM.

4.5 Experiments

4.5.1 Evaluation by Question Type

de Vries et al. (2017a) show that the “blind” version of the LSTM model performs better than the version receiving the visual features. This result is heavily dependent on the question type distribution in human-human dialogues. As we have seen, entity questions are a great proportion of the questions humans ask. The “blind” baseline model is facilitated in answering them, since it is given the category of the target object. Following Mazuecos et al. (2020), we evaluate models accuracy by question types. As we can see from Table 4.3, the higher overall accuracy reached by the “blind” LSTM model is indeed mostly due to the “entity” questions for which it reaches 94% (questions like: “is it a vehicle?”). As expected, when removing the category (V-LSTM) the accuracy on answering questions about entities decreases to a large degree, but the use of visual features helps the model to answer color questions better. The replacement of the LXMERT architecture, together with the use of positional aware embedding representations of the image, bring an important boost in the accuracy: LXMERT trained from scratch outperforms the LSTM based model on all types of questions. The pre-training phase further increases the performance in important ways.

	Absolute	Relational	Group
LSTM	76.4	67.1	63.3
V-LSTM	75.2	63.5	62.8
LXMERT-S	80.5	69.6	68.4
LXMERT	83.4	77.2	71.6

Table 4.4: Accuracy of the sub-type of spatial questions (successful games, questions assigned only one type)

4.5.2 Evaluation on Spatial Questions

Above we have seen that LXMERT outperforms the other models on the spatial questions. Our fine-grained classification sheds light on an interesting point: its main advantage comes from the *relational* questions (Table 4.4). *Absolute* questions require cross-modal attention only to align a word with its referent, whereas *relational* questions are more challenging: the model has to locate the regions corresponding to the two related words and understand the relation holding among them. The *group* questions may require “counting” skills that go beyond the scope of this chapter.

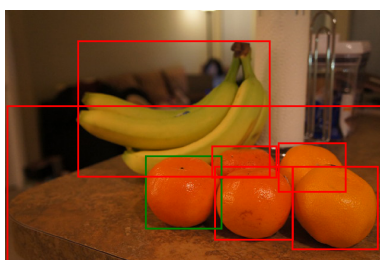
4.6 Qualitative Analysis

As a first step towards a deeper understanding of LXMERT performance, we use a linear logistic regression model for the task of predicting whether a question was answered correctly. In Shekhar et al. (2018) it has been shown that unsuccessful games contain more objects in the image than successful ones, and that the target size area is smaller. We use these two features as predictor variables together with the length of the question and the turn in which it was asked in the full dialogue. We observe that the number of objects in the image and the question turn play a significant role in predicting the model behaviour. This might be due to the fact that models do not receive the dialogue history as input. Below we run an error analysis based on the three spatial sub-type questions described above to check whether indeed this could be a source of error. After the error analysis, we study whether LXMERT uses its cross-modal attention differently across these three groups of questions.

4.6.1 Error Analysis

We did a manual error analysis of 20% of LXMERT errors on spatial questions. We tagged emergent error categories by following a qualitative annotation methodology. Below we describe our findings by classifying them in the three types of spatial questions that we consider throughout the chapter.

We found that absolute and group questions have more errors related to the missing dialogue history than relational questions even though we explicitly allow for relational questions that include anaphoric pronouns. For these two categories, around 50% of errors are related to missing dialogue history. Dialogue history dependency in the dataset is generally not lexicalized with explicit pronouns but left implicit through ellipsis (e.g. “in the middle?”). Figure 4.2 shows an example of this. Question 5 could be answered with “yes” if asked at the beginning of the dialogue (“middle” would refer to the middle of the image) but its answer is “no” due to history (“middle” refers to the middle of the group of oranges). In most of these dialogues, the category of the target is left implicit because it is established in previous questions (e.g., “orange”). But also other information is implicit. For example, “the last single one?” does not say that the search is evolving from right to left. In these cases, the meaning of the question is only correctly interpretable in the dialogue context.



Human question	Human answer
1. It is a fruit?	yes
2. It is the orange?	yes
3. One of them I suppose?	yes
4. Is it to our right?	no
5. In the middle?	no
6. The last single one?	yes

Figure 4.2: Sample image and dialogue from the GuessWhat?! dataset. The red boxes mark the objects involved in the questions, while the green box marks the actual referent. LXMERT incorrectly answers “yes” to question 5. LXMERT, like all Oracles, does not have access to the dialogue history. It probably interprets the question as “is the target in the middle of the picture?”. The image and dialogue illustrates the history dependence of questions.

History dependence, as illustrated in Figure 4.2, is hard to detect even for human anno-

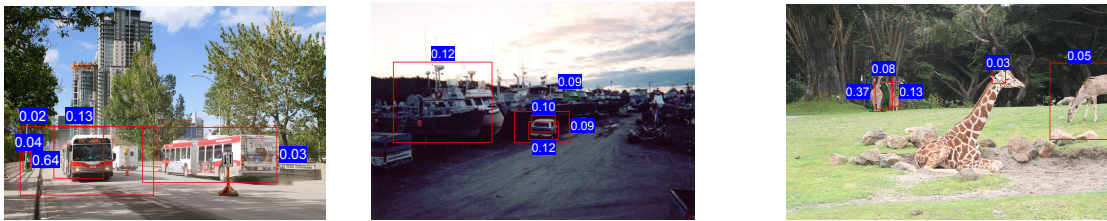
tators. Using the presence of the pronoun to detect whether a question needs the history in order to be properly answered, as it has been done in [Agarwal et al. \(2020\)](#), might be misleading. Our examples show that ellipses might create more context dependencies and that there are questions which could be apparently answered even when given in isolation but they would be answered differently based on the context they are in.

For absolute only questions, we found the following errors. Questions related to the z-axis of the picture (e.g. “is it in the background?”) seem to be harder for the model than those questions related to the x-axis of the picture (e.g. “is it on the left?”). The errors that do occur on the x-axis are either related to the fact that the dialogue history is necessary in order to interpret the question as in [Figure 4.2](#), or that the target is neither on the left nor on the right of the x-axis. In this dataset the adjective left and right behave as vague adjectives. Questions that include superlatives (e.g. “the rightmost book?”) cause many errors. As well as questions that combine two or more of these characteristics (e.g. “is it the animal at the very front on the left ?”). Finally, the ambiguity of the word “middle”, which could be used for any axis, seems to confuse the model.

For group questions, the second most frequent errors corresponds to questions grouping in one of the three axes. The term “row” is often used to group the target with other objects, especially when images are overcrowded with objects belonging to the same category. However, the term is an ambiguous one, as it can refer to any of the three axes and its meaning is often dependent on which interpretation is more salient in the image. Furthermore, inverse x-axis properties (e.g., “third girl from right?”) also seem to be problematic. Another frequent error type includes questions that require counting above three (e.g., “seventh bus from the left?”). People can immediately and precisely identify that an image contains 1, 2, 3 or 4 items by a simple glance, this ability is called subitizing ([Kaufman et al., 1949](#); [Piazza et al., 2002](#)). Identifying the quantity of a larger number of objects takes considerably longer and involves counting for humans. It seems models such as LXMERT are able to do subitizing, but not counting. Other problematic group questions are multi-type ones, for instance belonging also to the relational type (e.g., “are there two of them on the branch?”); and questions using entities outside the image as reference, such as the viewers (e.g., “is it in the first room closer to us?”).

Layer	Absolute	Relational	Group
0	3.9	4.1	3.3
1	4.2	4.6	4.1
2	3.8	4.5	4.0
3	3.7	4.0	3.7
4	1.3	2.2	1.9

Table 4.5: Language to Vision attention in LXMERT: Number of regions of the image considered salient in the last layer from the CLS token – viz. regions with an attention value higher than the 0.05 threshold.



is it the bus on the left? No is it the boat next to a car? No is it one of the two in the back? Yes

Figure 4.3: Attentions from the CLS: in absolute questions attention is mostly on the only object the question refers to (the left bus, 0.13) and the target object (0.64) (**left**); in the relational questions attentions spread between the two related objects (car and boat, 0.12 each) and the target object (the boat on the back, 0.9) (**middle**); in the group questions attentions goes to the entity of the referred group (0.08 and 0.13) and the target (0.37) (**right**).

For relational questions we find that a source of errors is when the target and the landmark bounding boxes overlap or one is included in the other (“is it the clock behind the person?”). Also when the landmark is a part of another object instead of being an object with well delimited borders the model seems to get confused (“is it under his feet?”). Questions that include non projective prepositions seem harder (“is it the person near the bicycle?”) than those whose prepositions indicate the direction of the relation. Another source of errors are questions in which the landmark is large and no clear borders are visible (“is it on the water?”). Finally, those questions that require OCR (optical character recognition) are problematic (“does it have words on it?”).

4.6.2 LXMERT’s Attention

Here we aim to understand how LXMERT uses attention mechanisms to answer spatial questions. We focus our analysis on the cross-attention layers from language to vision.

Recall that, in our adaptation of LXMERT to the Oracle task, the crop of the target is given as the 36th visual embedding together with the most salient regions of the image detected by Faster R-CNN. We are interested in understanding how it exploits the target visual representation to guide attention.

The entropy of the attention maps shows that the model in the first attention layers distributes attention across all regions (its entropy is close to the maximum possible level), at layer 2 it learns to focus its attention on some regions of the image and on the crop of the target. Finally, at the last layer, the attention on the CLS (the embedding given to the classifier to select the answer) reveals an interesting difference among question types: the number of regions considered salient in the absolute questions is lower than the one of salient regions in the group and relational questions. Table 4.5 reports the numbers of regions with an attention value higher than 0.05.³ We have used different thresholds to compute the number of top-valued regions and the same pattern emerges. From a manual inspection, we have seen that the higher number of salient regions in the relational questions often is due to the fact that they refer to more candidate objects, differently from the absolute ones which usually refer to fewer or even just one object.

Figure 4.3 illustrates how LXMERT uses its attention in three sub-type of spatial questions. As we can see, when it interprets relational questions involving two objects, it “looks” both at the target (the boat) and the landmark (the car); in the example it answers the question negatively since the target of the game is the boat marked by the green box and not the one to which the question refers to. Similarly, when interpreting a group question, it looks at the referred group (the two giraffes); in the example it answers the question positively since the target of the game is indeed within the referred group. By looking at the attention maps, we noticed that interesting patterns emerge when looking at the attentions from the CLS token (Figure 4.3 marks the regions considered more salient from the CLS token). Other tokens put attention mostly or only on the target object region.

³If the attention is equally distributed among all the 36 regions, their attention value would be 0.02 (viz. 1/36).

4.7 Conclusion

In this chapter we tackle the problem of grounding spatial questions in the GuessWhat?! visual dialogue game. We adapt LXMERT to play the role of the Oracle of the Guess-What?! game reaching an overall accuracy of 82.21%. This result outperforms the widely used baseline model by 6.27%. The gain is even higher for spatial questions, where LXMERT outperforms the baseline by 9.70%. In order to perform an in-depth analysis, we classify spatial questions into three sub-types and use this classification to annotate the subset of spatial questions in the GuessWhat?! test set. The fine-grained evaluation shows that the hardest spatial questions are the relational and group ones. We perform an in-depth analysis of LXMERT cross-modal attention and an qualitative analysis of the errors on each question sub-type. First of all, we find out that LXMERT puts attention on more regions when processing relational questions compared to absolute and group questions. Secondly, the qualitative analysis highlights the importance of having access to the dialogue history in order to answer some spatial questions. We leave this for future work.

4.8 Summary

So far, we discovered that:

- + Pre-Trained Transformers bring a huge improvement in the task of answering spatial questions about a target object in an image;
- + Pre-Trained Transformers pinpoint the most important regions in the image;
- + Pre-Training makes the biggest difference with respect to the other models;
- Further improvements can be obtained for group questions.

Chapter 5

Pre-Trained Transformers Encoding Positive and Negative Answers

Selecting the target object in a referential visual dialogue requires to understand each utterance in the dialogue at a fine-grained level. In the case of GuessWhat?!, this involves identifying questions and answers in each utterance, and exploiting the information that they convey. Models should leverage both positive and negative answers. We take GuessWhat?! as test-bed and evaluate to which extent guessers based on pre-trained Transformers profit from positively and negatively answered polar questions. Moreover, in order to get a better grasp of models' results, we select a controlled sample of games and run a crowd-sourcing experiment with humans subjects. We evaluate models and humans against the same settings and use the comparison to better interpret the models' results. We show that pre-Trained Transformers are able to understand the structure of a dialogue. However, while humans profit from negatively answered questions to solve the task, models struggle in grounding negation, and some of them barely use it. It is worth noting that when the language signal is poorly informative, visual features help encoding the negative information. Finally, the experiments with human subjects put us in the position of comparing humans and models' predictions and get a grasp about which models make errors that are more human-like and as such more plausible.

5.1 Introduction

Negation is often neglected by computational studies of natural language understanding, in particular when using the successful neural network models. Very recently, a series of work have highlighted that negation is under-represented in existing natural language inference benchmarks (Hossain et al., 2020b) and that Pretrained Language Models have difficulty distinguishing a sentence from its negated form in fill-in-the-blank tests (Kassner and Schütze, 2020). This weakness of Language Models could have a strong impact on their success in real-life applications. For instance, Hossain et al. (2020a) show that the lack of a proper understanding of negation is an important source of error in machine translation and similarly, it would impact the quality of other applications based on natural language understanding, such as text summarization or personal assistants for health care or other uses. A recent contribution of AI to the society is the development of visual dialogue systems built on Pretrained Language Models. Clearly, they are an important tool for instance as personal assistants of visually impaired people (Gurari et al., 2018a), but again their impressive achievements would be vanished if they fail to distinguish negative and affirmative information.

Admittedly, modelling negation is an ambitious goal, and even humans have a harder time understanding negative sentences than positive ones (Clark and Chase, 1972; Carpenter and Just, 1975). However, it has been shown that the presence of supportive context mitigates the processing cost of negation. In particular, this happens within dialogues (Dale and Duran, 2011), and when a visual context is given (Nordmeyer and Frank, 2014). Based on these findings, we argue that Visual Dialogues are a good starting point for making progress towards the ambitious but crucial goal of developing neural network models that can understand negation.

Visual Dialogues can be chit-chat (Das et al., 2017a) or task-oriented (de Vries et al., 2017b; Ilinykh et al., 2019b; Haber et al., 2019; Ilinykh et al., 2019a). Task-oriented

This chapter describes the work by (Testoni et al., 2021).

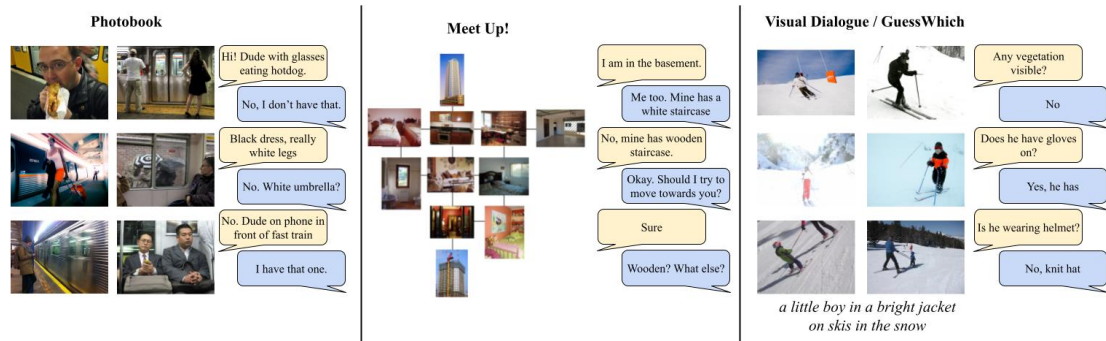


Figure 5.1: Examples of dialogues from two asymmetric and partially observable visual dialogue data (PhotoBook and Meet Up! (Haber et al., 2019; Ilinykh et al., 2019a)) and a symmetric visual dialogue in which the answerer sees the image and the questioner does not see it (Das et al., 2017a; Chattopadhyay et al., 2017). For all datasets, we selected exchanges containing negation, the focus of our study.

dialogues are easier to evaluate since their performance can be judged in terms of their task-success, hence we focus on this type of dialogues which can be further divided as following: the two agents can have access to the same visual information (de Vries et al., 2017b), share only part of it (Haber et al., 2019; Ilinykh et al., 2019a) or only one agent has access to the image (Chattopadhyay et al., 2017). Moreover, dialogues can be symmetric (Haber et al., 2019), or asymmetric, with one agent asking questions and the other answering it (de Vries et al., 2017b; Das et al., 2017a; Chattopadhyay et al., 2017). Finally, the dialogue turns can contain different speech acts (Ilinykh et al., 2019b; Haber et al., 2019; Ilinykh et al., 2019a) or only question answer pairs (de Vries et al., 2017b; Das et al., 2017a; Chattopadhyay et al., 2017). The differences between the various type of dialogues are illustrated in Figure 5.1. As we can see symmetric games with partially observable data (PhotoBook and Meet up! (Haber et al., 2019; Ilinykh et al., 2019a)) solicitate more complex exchanges than symmetric ones (Visual Dialogue, GuessWhich – the referential game built from it (Das et al., 2017a; Chattopadhyay et al., 2017), and GuessWhat?! (de Vries et al., 2017b) – the latter is illustrated in Figure 5.2). Given the difficulty negation poses to models, we take the scenario which is less complex from a dialogue perspective and in which questions are always grounded in the image: the one in which agents have access to the same visual information, only one agent can ask questions, and the questions are all of the same type. Hence, we take GuessWhat?! as case-study and focus on the referential grounded guessing task: a Guesser receives an asymmetric dialogue, consisting of Yes/No-questions over an image, a list of candi-

dates and has to guess the target object the dialogue is about. In this setting, negation is heavily present as the answer to a binary question. As such it functions as a pointer to the alternative set of the negated expression; in other words it should be interpreted as pointing to the set of all the candidates objects which do not have the queried property.

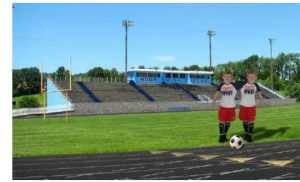
GuessWhat?! dialogues have been collected by letting two humans play the game. As illustrated in Figure 5.2, such dialogues are quite simple: a sequence of rather short questions answered by “Yes” or “No” containing on average 30.1 (SD \pm 17.6) tokens per dialogue. The dialogue length differs across the games since the questioner decides when he/she can stop asking questions and is ready to guess the target. To evaluate the extent models understand negatively answered questions, we take the human dialogues as input to the guesser. We select successful games, in other words those dialogues in which human players have succeeded in guessing the target object at the end of the game. We conjecture that within these dialogues a crucial role is played by the last turn whose role is to create a singleton alternative set and that this goal is achieved differently when the question is answered positively or negatively. In the former case, the question tends to almost fully describe the target object, whereas in the latter case it conclusively identifies the target object by excluding those candidates which most likely are not the target (Figure 5.2). To validate this conjecture, we run an online experiment with humans which set the ground for better evaluating the results obtained by models. We let humans and computational models perform the same task on the same controlled sample set. We compare encoders with respect to the architecture (Recurrent Neural Networks vs. Transformers), the input modalities (only language vs. language and vision) and the model background knowledge (trained from scratch vs. pre-trained and then fine-tuned on the downstream task). Our analysis shows that:

- While humans profit from negatively answered questions to solve the task, models struggle in grounding negation, and some of them barely use it;
- In No-turns, when the language signal is poorly informative, visual features help in processing the QA pair.

We hope that these results will stimulate more work on the processing of (grounded)



Questioner	Oracle
1. Is it on a wooden surface?	Yes
2. Is it red?	No
3. Is it white?	No
4. Is it a scissor?	Yes
5. Is it the scissor on the left of the picture?	Yes



Questioner	Oracle
Q1. Is it an object?	No
Q2. Is it a person?	Yes
Q3. Does he have his right arm on the other's shoulder?	No

Figure 5.2: Two samples of GuessWhat?! human dialogues ending with a positive (left) and a negative (right) turn.

negation and that the data we collected through our online experiment and its annotation will be a valuable contribution to such research direction.¹

5.2 Related Work

Scrutinizing Visual Dialogue Encoding Sankar et al. (2019) study how neural dialogue models encode the dialogue history when generating the next utterance. They show that neither recurrent nor transformer based architectures are sensitive to perturbations in the dialogue history and that Transformers are less sensitive than recurrent models to perturbations that scramble the conversational structure; furthermore, their findings suggest that models enhanced with attention mechanisms use more information from the dialogue history than their vanilla counterpart. We follow them in the choice of the architectures we compare, but we change the focus of the analysis by studying whether the polarity of the answer (Yes vs. No) affects the encoding of the information provided by the question-answer pair.

Kaushik and Lipton (2018) show that in many reading comprehension datasets, that presumably require the combination of both questions and passages to predict the correct answer, models can achieve quite a good accuracy by using only part of the information

¹https://github.com/albertotestoni/annotation_human_gw

provided. Similarly to this work, we investigate how much models use the questions as well as the answers, provided by the Oracle, to select the target object among the possible candidates.

As shown in Chapter 3 (Greco et al., 2020), pre-trained transformers detect salient information in the dialogue history independently of the position in which it occurs. We build on their study to dive into how encoders represent positively vs. negatively answered questions within a visual dialogue.

SOTA LSTM-based Models on GuessWhat?! After the introduction of the supervised baseline model (de Vries et al., 2017b), several models have been proposed. Zhao and Tresp (2018) have used attention mechanisms based on Memory Networks (Sukhbaatar et al., 2015). Shekhar et al. (2019) have proposed a model that is jointly trained to ask questions and guess the target. Building on the supervised learning step, all these models have been further trained with either some form of reinforcement learning (Zhang et al., 2018b; Zhao and Tresp, 2018; Yang et al., 2019; Pang and Wang, 2020) or cooperative learning (Shekhar et al., 2019; Pang and Wang, 2020); this two-step process has been shown to reach higher task success than the supervised approach. Since our focus is on the Guesser and we are evaluating it on human dialogues, we will compare models that have undergone only the supervised training step.

Transformer-based Models The last years have seen the increasing popularity of transformer-based models pre-trained on several tasks to learn task-agnostic multimodal representations (Li et al., 2019; Lu et al., 2019a; Tan and Bansal, 2019; Chen et al., 2019a; Su et al., 2020; and Nan Duan et al., 2020). ViLBERT (Lu et al., 2019a) has been recently extended by means of multi-task training involving 12 datasets which include GuessWhat?! (Lu et al., 2020) and has been fine-tuned to play the Answerer of VisDial (Murahari et al., 2019a). In Chapter 3 (Greco et al., 2020) we have adapted the pre-trained transformer, LXMERT (Tan and Bansal, 2019), to the GuessWhat?! guessing task. Given the high accuracy achieved, we choose LXMERT as pre-trained transformer.

Visually Grounded Negation Negation was already listed by Winograd among the linguistic phenomena a Grounded Conversational System should be able to interpret (Winograd, 1972). Significant progress has been obtained in the development of conversational systems based on neural network architecture; however, little is known about how these models interpret negation. Nordmeyer and Frank (2014) show that processing negation can be easier for humans if a visual context creates pragmatic expectations that motivate its use. However, it is unknown whether this holds for multimodal models. Suhr et al. (2019) show that SOTA models tested on visual reasoning often fail in properly grounding negative utterances. Gokhale et al. (2020) show that models have harder time in answering visual questions containing negation. Both studies look at negation as a logical operation, it reverses the truth value of the negated utterance. However, Oxford (2002) show that humans often use negation not as a logical operator but rather as a way to create an alternative set of the negated expressions. This is exactly the role of the negative answer in the GuessWhat?! game. We are not aware of any study on Visual Dialogue that have tackled this issue.

5.3 Task and Dataset

In this chapter, we run an in-depth analysis on how models integrate Yes/No answers into the question to solve the GuessWhat?! guessing task. We run a comparative analysis to evaluate the role of language priors and visual grounding, and we run a crowdsourcing experiment with subjects on a controlled sample of the games. Using a controlled sample set and knowing about humans' performance give us a better way to interpret the results obtained by the models on the full test set. Below we describe the task and training/validation set and the test sets we use through out the experiments.

We rely on GuessWhat?!, a referential game involving a dialogue about images previously described in section 2.5.2. Our focus is on multimodal encoding of a visual dialogue we focus on the following visual guessing task: given a human dialogue, consisting of Yes/No questions and their answers, an image and a list of possible candidate objects, the agent has to select the object the dialogue is about. In 2.5.2, it has been

shown that human dialogue length is a good proxy of the guessing task difficulty, where length is measured in terms of number of turns. Indeed, in shorter dialogues the area of the target object is bigger than in longer dialogues, and in short dialogues the target object is quite often a “person” – the most common target category in the dataset; moreover, the number of distractors in longer dialogues is much higher. For instance in Figure 5.2 the dialogue on the left is of length 5 (it consists of five turns) whereas the one on the right is of length 3. In the following, we use “turn” to refer to the position (of just the question or the answer or of the QA pair) within the dialogue.

Full dataset We evaluate models using human dialogues, selecting only the games on which human players have succeeded finding the target and contain at most 10 turns (total number of dialogues used: 90K in training and around 18K both in validation and testing). Dialogues contain on average 4.5 Question-Answer (QA) pairs, the vocabulary consists of 4901 words, and games have on average 8 candidates.² The answer distribution is the following: 52.2% No, 45.6% Yes, and 2.2% N/A (not applicable). We divide the full test set into games whose dialogue ends in a Yes- vs. in a No-turn and obtain the Yes-set and No-set, whose statistics are reported in Table 5.1. As we can see, the two sets contain dialogues of the same average length, and similar number of candidate objects, hence their games are expected to be of similar difficulty. The last turns in these two subsets are expected to play a rather different role (as illustrated by the example in Figure 5.2): a Yes-question in the last turn is rather informative on its own, whereas a last turn answered negatively quite often needs the information gathered in the previous turns to be informative. On the other hand, we should note that last turns containing a negative answer are expected to be rather informative together with the dialogue history to guess the target. Hence, they are an interesting test-bed for our research question.

Controlled Sample To compare models’ results against humans’ ones, we run an annotation experiment on a sample of games we carefully select. We consider dialogues consisting of 4- and 6-turns, and select those containing an equal number of Yes/No answers. Moreover, to control for the level of difficulty of the game, we select only games which have a maximum of 10 candidates. We obtain a subset with a balanced overall distribution of the two types of polar answers; it contains 1491 games, of which 1327

²The dataset of human dialogues is available at <https://guesswhat.ai/download>.

	Nr. Games	Av. Dialogue length	Av. nr candidates
Full test set	18840	4.5	8
Yes-set	16366	4.5	8
No-set	2350	4.5	7.8
Controlled Sample	300	4.5	6.1
Yes-set	150	4.5	6.1
No-set	150	4.3	6.1

Table 5.1: Statistics on the full test set and on the Controlled test set; both divided into the Yes- (resp. No-) subsets obtained by selecting only dialogues with a positively (resp. negatively) answered question in the last turn.

(resp. 164) contain in the last turn a question answered positively (resp. negatively). From these games, we randomly select 300 games (image, target) from the Yes- and No- test sets (150 each). In this way, we obtain a subset balanced also with respect of the polarity of the last question. We believe games in this sample set are equally difficult, considering the criteria discussed above.

5.4 Models

Following Chapter 3 (Greco et al., 2020), all the guesser models we evaluate share the skeleton illustrated in Figure 5.3: an encoder paired with a Guesser module. For the latter, all models use the module proposed in de Vries et al. (2017b). Candidate objects are represented by the embeddings obtained via a Multi-Layer Perceptron (MLP) starting from the category and spatial coordinates of each candidate object. The representations so obtained are used to compute dot products with the hidden dialogue state produced by an encoder. The scores of each candidate object are given to a softmax classifier to choose the object with the highest probability. The Guesser is trained in a supervised learning paradigm, receiving the complete human dialogue history at once. The models we compare differ in how the hidden dialogue state is computed. We compare LSTM vs. Transformers when receiving only the language input (Language-only, henceforth, Blind models) or both the language and the visual input (Multimodal, henceforth, MM models).

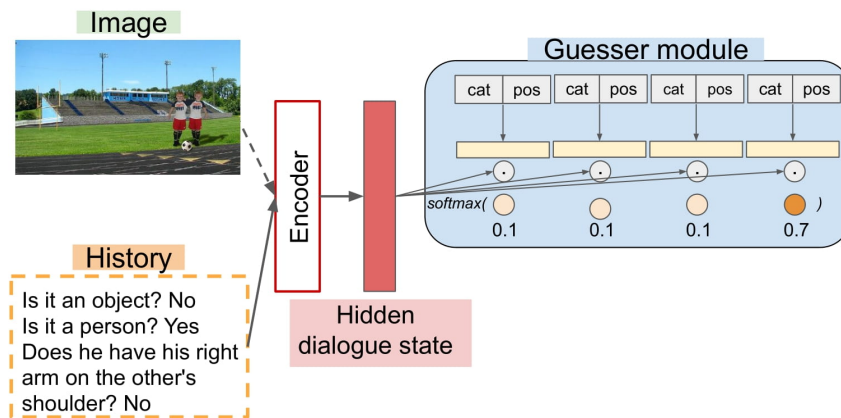


Figure 5.3: Shared Encoder-Guesser skeleton. The Guesser receives the category labels (e.g., “bottle”) and the spatial coordinates (pos) of each candidate object. Multimodal encoders receive both the image and the dialogue history, whereas blind models receive only the latter.

5.4.1 Language-only Encoders

LSTM As in [de Vries et al. \(2017b\)](#), the representations of the candidates are fused with the last hidden state obtained by an LSTM which processes only the dialogue history.

RoBERTa In the architecture of the model described above, we replace the LSTM with RoBERTa ([Liu et al., 2019b](#)).³ We use RoBERTa_{BASE} which has been pre-trained on 160GB of English text trained for 500K steps to perform masked language modeling. RoBERTa was pre-trained on several text corpora containing rather long utterances: BookCorpus ([Zhu et al., 2015b](#)) + English Wikipedia (as the original BERT model), CC-NEWS ([Nagel, 2016](#)), OPENWEBTEXT ([Gokaslan and Cohen, 2019](#)), and STORIES ([Trinh and Le, 2018](#)). It has 12 self-attention layers with 12 heads each. It uses three special tokens, namely CLS, which is taken to be the representation of the given sequence, SEP, which separates sequences, and EOS, which denotes the end of the input. We give the output corresponding to the CLS token to a linear layer and a *tanh* activation function to obtain the hidden state which is given to the Guesser. To study the impact of the pre-training phase, we have compared the publicly available pre-trained model, which we fine-tuned on GuessWhat?! (**RoBERTa**), against its counterpart trained from scratch only on the game (**RoBERTa-S**).

³We have also tried BERT, but we obtained a higher accuracy with RoBERTa.

5.4.2 Multimodal Encoders

V-LSTM We enhance the LSTM model described above with the visual modality by concatenating the linguistic and visual representation and scaling its result with an MLP; the result is passed through a linear layer and a *tanh* activation function to obtain the hidden state which is used as input for the Guesser module. We use a frozen ResNet-152 pre-trained on ImageNet (He et al., 2016) to extract the visual vectors.

LXMERT In order to evaluate the performance of a universal multimodal encoder, we employ LXMERT (Tan and Bansal, 2019). It represents an image by the set of position-aware object embeddings for the 36 most salient regions detected by a Faster R-CNN and it processes the text input by position-aware randomly-initialized word embeddings. LXMERT is pre-trained on datasets containing rather short utterances: MSCOCO (Lin et al., 2014a), Visual Genome (Krishna et al., 2017a), VQA v2.0 (Antol et al., 2015a), GQA balanced version (Hudson and Manning, 2019), and VG-QA (Zhu et al., 2016). Both the visual and linguistic representations are processed by a specialized transformer encoder based on self-attention layers; their outputs are then processed by a cross-modality encoder that, through a cross-attention mechanism, generates representations of the single modality (language and visual output) enhanced with the other modality as well as their joint representation (cross-modality output). Like RoBERTa, LXMERT uses the special tokens CLS and SEP. Differently from RoBERTa, LXMERT uses the special token SEP both to separate sequences and to denote the end of the textual input. LXMERT has been pre-trained on five tasks.⁴ It has 19 attention layers: 9 and 5 self-attention layers in the language and visual encoders, respectively and 5 cross-attention layers. We process the output corresponding to the CLS token as in RoBERTa. Similarly, we consider both the pre-trained version (**LXMERT**) and the one trained from scratch (**LXMERT-S**).⁵

⁴Masked cross-modality language modeling, masked object prediction via RoI-feature regression, masked object prediction via detected-label classification, cross-modality matching, and image question answering.

⁵We use the code available from <https://github.com/claudiogreco/aixia2021>

5.5 Experiments on the Full Test set

We aim to understand whether models encode Yes/No answers and properly integrate them into the question. If answers play a role in the performance of the models in guessing the target object, removing them from the dialogues should cause a drop in the task accuracy. Following this conjecture, we evaluate models (at test time, without additional training) when receiving only the questions from the dialogues (without the answers). Moreover, as commented above, the last turn in the Yes-set vs. No-set is expected to play a rather different role. In particular, already alone a positively answered question in the last turn is expected to be rather informative whereas a last turn answered negatively is not. On the other hand, last turns containing a negative answer are expected to enrich the dialogue history and help to guess the target. Hence, in the following, we evaluate models aiming to understand the role of the last turn.

5.5.1 Accuracy results

Only Questions We evaluate models when receiving dialogues containing only the questions.⁶ As expected, all models show an important drop as we can see from Table 5.2. Blind models have higher accuracy than the multimodal counterpart when receiving only the question, maybe because during training they learn to exploit the language surface more. Moreover, the pre-training phase helps to exploit the keywords in the questions as shown by the difference between the pre-trained and from scratch versions of both transformer based models. These results show that all models take the answers into account to some extent, and thus it is important to study their impact on the performance of the models.

Dialogues with a Yes- vs No- answer in the last turn We now investigate how the polarity of the final answer in the dialogue affects the performance in the guessing task. Models reach a rather lower accuracy on the No-set, suggesting that models have harder time interpreting dialogues ending with a negative answer (Table 5.2). Differently from what one would expect, it seems the pre-trained transformer that does not have access

⁶We replaced all the answers with the “unknown” [UNK] token.

		Full dialogue all games	Only Q all games	Full dialogue	
				Yes-set	No-set
BLIND	Random	12.5	12.5	16.4	16.4
	LSTM	64.7	47.9	67.0	49.0
	RoBERTa-S	64.2	43.7	66.6	48.1
	RoBERTa	67.9	51.7	69.6	54.5
MM	V-LSTM	64.5	46.2	67.0	48.3
	LXMERT-S	64.4	32.0	66.6	49.5
	LXMERT	69.2	44.8	71.9	50.9

Table 5.2: **Full test set:** Task Accuracy obtained by models when receiving: a) only the questions (Only Q); b) the full dialogue in the Yes-set vs. No-set, viz. games ending with a Yes-turn vs. a No-turn. All differences between RoBERTa and LXMERT are statistically significant.

	Yes-set		No-set	
	W/o Last	Last	W/o Last	Last
LSTM	48.3	51.8	39.9	24.5
RoBERTa-S	49.8	50.7	39.6	21.8
RoBERTa	53.5	55.6	42.5*	23.3
V-LSTM	48.6	47.3	37.8	20.7
LXMERT-S	48.4	51.7	41.0	22.2
LXMERT	49.9	61.2	41.9*	26.6

Table 5.3: **Full test set:** Accuracy comparison when giving to the model the dialogue without the last turn (W/o Last) or with only the last turn (Last). (The * marks RoBERTa’s and LXMERT’s scores whose differences are statistically not significant.)

to the visual representation of the “alternative set” (RoBERTa) performs better than the multimodal model, LXMERT, in the challenging No-set games. It is not clear, however, where the advantage of RoBERTa comes from. Hence, in the next section, we aim to understand these results better by using the controlled sample and comparing models against the humans’ performance, with a particular focus on the role of the last dialogue turn.

The role of the last turn To analyze the role of the last turn, we compute models’ accuracy when receiving the dialogues without the last turn or with only the last turn. The drop obtained from the setting in which models have access to the full dialogue quantifies the role of the last turn. First of all, as shown in Table 5.3, when removing the last turn in the Yes-set, LXMERT has a higher drop in accuracy than RoBERTa: -22.0% (from 71.9% to 49.9%) vs. -16.1% (from 69.6% to 53.5%); the fact that LXMERT relies

on the last turn a lot might be due to LXMERT having harder time than RoBERTa in encoding the dialogue history, as observed in Chapter 3 (Greco et al., 2020). When only the last turn is provided, LXMERT profits from the pre-training phase more than RoBERTa. Recall that LXMERT has seen shorter text than RoBERTa during training, e.g. MS-COCO captions vs. Wikipedia text. This difference could be behind such results. In the No-set, LXMERT processes the last turn better than RoBERTa (it reaches 26.6 accuracy when receiving only the last turn, +3.3 than RoBERTa), but again it has more difficulty in integrating such information with that gathered through the dialogue history (it scores -3.4% than RoBERTa when receiving the full dialogue). Finally, as expected, when receiving only the last turn, models obtain a high accuracy when the answer is positive (Yes-set) and are near to chance level when it is negative (No-set). Interestingly, in the No-set, RoBERTa and LXMERT have a rather similar accuracy when the last turn is not given and LXMERT does slightly better than the language encoder when receiving only the last turn. These results suggest that the advantage of RoBERTa over LXMERT highlighted in Table 5.2 is due to a better processing of the whole dialogue history, while LXMERT exploits better shorter sequences such as the last turn taken individually in the No-Set (Table 5.3).

Tests of Statistical Significance To validate our findings about the comparison between RoBERTa and LXMERT, we have run the McNemar’s test with a significance level of 0.05. We use an asterisk to signal scores whose differences is not significant (Table 5.2 and 5.3).

5.5.2 Guesser’s Probability Distribution

We now analyze how the guesser module assigns probabilities to the target object across the turns to understand better the role of positive and negative answers at a more fine-grained level. We compute how the probability assigned by the Guesser to the target object $P(o)$ changes after each turn ($P(o)_{T_{i+i}} - P(o)_{T_i}$) and compare turns T_i with a Yes, No or N/A answer. We expect it is easier to use the Yes-turns than the No ones, but we hope models are able to benefit from the questions answered negatively more than those answered by N/A. Moreover, we focus on the games in which the Guesser succeeds to

	All games			All successful games		
	Full dialogue history			Last turn		
	$T_i : Yes$	$T_i : No$	$T_i : N/A$	$T_i : Yes$	$T_i : No$	$T_i : N/A$
LSTM	14.5	2.9	2.3	26.3	16.2	6.3
RoBERTa-S	12.7	3.5	1.9	24.6	16.4	1.1
RoBERTa	12.3	5.9	1.4	22.9	18.8	1.1
V-LSTM	14.0	3.1	2.9	23.7	13.7	6.7
LXMERT-S	12.3	4.4	2.1	24.8	19.3	0.7
LXMERT	16.4	4.1	1.4	30.0	24.9	3.2

Table 5.4: Change across consecutive turns in the probability assigned to the target after Yes- vs. No- vs. N/A-turns, i.e., $P(o)_{T_{i+1}} - P(o)_{T_i}$ (full dialogue history in the full test set) and before/after the last turn (Last turn in games on which the model has succeeded).

select the target object, and quantify the effect of the last turn on the probability assigned to the target. We expect the change in the last turn of the No-set to be much higher than No-turns in average, whereas this should not happen with last turn in the Yes-set.

Although the average probability assigned to the target is similar before a Yes-turn and a No-tun for all models⁷, questions answered with Yes bring a much higher increase of probability than questions answered with No – which for LSTM have on average the same impact as those answered by N/A (2.9 vs. 2.3) (Table 5.4).⁸ Again, RoBERTa is the model that seems to profit of the negative turn more: the probability the guesser assigns to the target object after a No-turn increases of 5.9 vs. 4.1 when using LXMERT as encoder. However, when we focus on the last turn (Table 5.4-right), LXMERT is the model for which the negative answer brings a higher increase to the target object. In the following, by zooming into the controlled sample we aim to get a more accurate comparison of models with respect to the specific issue of how they encode negatively answered questions.

⁷The difference is lower than 10%.

⁸In Table 5.4 we report the results for all the dialogues. Similar patterns have been seen when comparing the models on games with a given number of candidate objects.

5.5.3 Summary

In short, the experiments run so far show that all models take the answer of the asymmetric GuessWhat?! dialogues into account. The pre-trained encoders are the best models over all games and are on par with one another in processing positively answered questions. But, the results on the Yes-set when removing the last turn or when giving only the last turn shows that LXMERT profits from the last Yes-turn more than RoBERTa. We conjecture this is due to the fact that LXMERT has a harder time encoding the dialogue history. The overall accuracy obtained on the No-set suggests that RoBERTa encodes the negatively answered questions better than LXMERT. However, an in-depth analysis of the Guesser probability distribution shows that the Guesser profits from the last turn in the No-set more when it is based on LXMERT than on RoBERTa. From the analyses presented so far, it emerges that the models we considered have different strengths and weaknesses, depending on many factors. To establish an upper-bound for models' performance and to assess the severity of the errors made by the models, in the following we present an in-depth analysis we carried out with human annotators playing the same guessing task of the models.

5.6 Controlled Sample: Humans and Models

In order to interpret models' performance on encoding Yes/No-turns, we evaluated humans' performance on the controlled games sample described in Section 5.3. These results set an upper-bound for model performance, and give us a powerful tool to better scrutinize our results.

5.6.1 Experiments and Results with Human Annotators

We asked human annotators to perform the GuessWhat?! guessing task on a controlled sample of test set games. Similarly to what discussed in Section 5.5, we evaluate several settings: we provide annotators with the full dialogue, the dialogue without the last

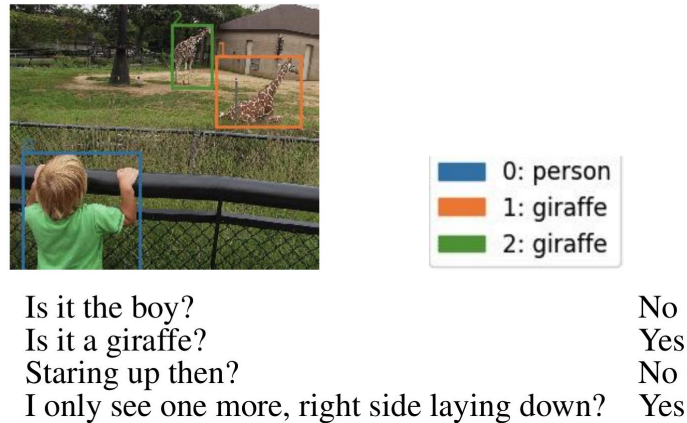


Figure 5.4: Prolific interface: Humans were given a dialogue, an image with colored bounding boxes, and a numbered list of candidates with colors matching those of the bounding boxes. They had to use the keyboard device to choose the target.

turns, or only the last turn. Moreover, to check the average informativeness of Yes- No- turns, we add the setting in which we remove from the dialogues all turns of the same polarity.

Data collection Through Prolific,⁹ we collected complete annotations from 48 subjects who were paid Euro 8.27/hour. Each participant annotated 75 games from one of the four settings. In total, we have collected 3600 human answers. Each setting has received annotation from 3 participants. Participants were asked to be native English speakers.

Participants were given an image with bounding boxes associated with each candidate object, together with a progressive ID number, as illustrated in Figure 5.4. They express their guess by pressing on the device’s keyboard the number corresponding to the chosen object. Before starting the experiment, they were shown three trial games for which the correct answer was displayed in case the annotator chose the wrong target. We added two control games in each setting, i.e., games with a full dialogue history and few candidate objects. Participants were told there were control games and that they would have been excluded from the data collection in case the wrong answer was given for those games. Only one annotator wrongly guessed the control games and was therefore excluded. We recorded the time taken by each participant to complete the experiment. On average, humans took 12.23 seconds for each datapoint in the group A (removing turns), 15.55 seconds for group B (without last turn), 10.52 seconds for

⁹<https://www.prolific.co/>

group C (only last turn), and finally 20.26 for group D (full dialogue). We found no statistically significant correlation between the time taken to guess the target and the success in solving the task.

Tests of Statistical Significance As we did in the previous section, we validate the accuracy results by running a McNemar’s test with a significance level of 0.05 (Table 5.5 and 5.7). Table 5.6 reports the times taken by humans to play games belonging to the different groups we have analyzed. The differences within groups are not normally distributed – Shapiro–Wilk test. Hence, to check the validity of such comparisons we have run a Wilcoxon rank-sum statistic for two samples using 0.05 as significance level. Again, we use asterisks to signal the results whose difference is not statistically significant.

Results with Humans As mentioned above, we focus on games on which human players have been successful in guessing the target object. It has to be noted that during the GuessWhat?! data collection, each game was played only once and the target object was guessed by the same player who asked the questions. Hence we do not know whether the same dialogue-image would be equally informative for another player to succeed in the game neither we know the level of uncertainty behind the choice made by the successful player. With these questions in mind, in Table 5.5 we report the accuracy obtained by humans in our controlled experiment by considering a game successfully solved if (a) at least one participant correctly identifies the target object among the list of candidates (the typical GuessWhat?! accuracy evaluation setting, modulo the fact that in our case the questions are already asked) and (b) at least two participants guess the target correctly (the most standard and solid evaluation); we refer to these two accuracy metrics as minority (MIN) and majority (MAJ) schema, respectively.

Given that we are working with games on which GuessWhat?! human players succeed guessing the target, the fact we do not obtain 100% accuracy in the group D (complete dialogues) is by itself interesting. The difference between the two schema shows that, also in the games successfully solved by human players in the GuessWhat?! dataset, there is a margin of uncertainty. As we see from the Table 5.5 (Group D, full dialogue), 98% of the games ending with a Yes-turn could be guessed by at least one participant

A) Removing turns			B) W/o last				C) only last		D) Full dialogue	
MAJ	Only Yes	66.00	MAJ	Yes-set	75.33	71.33	86.67*			
	Only No	46.00		No-set	49.33	30.67	80.67*			
MIN	Only Yes	80.67	MIN	Yes-set	92.00	88.00	98.00			
	Only No	72.67		No-set	64.67	58.00	90.00			

Table 5.5: Humans’ performance on controlled sample: percentage of games guessed correctly by at least two participants (MAJ) vs. by at least one participant (MIN). (* not significant)

Group	Description	Average time/token (s)
A	only yes turns	0.45
A	only no turns	0.57
B	without last (yes)	0.94*
B	without last (no)	0.79*
C	only last (yes)	1.20
C	only last (no)	2.53
D	full dial ending with yes	0.72
D	full dial ending with no	0.85

Table 5.6: Average time (seconds) taken by humans to solve games belonging to the different groups analyzed. Normalized with respect to the number of token in the text; only successful games are considered. (* not significant)

(minority schema) whereas 86.67% of them were guessed correctly by at least two participants (majority schema). Games ending with a No-turn are more difficult: 90% (resp. 80.67%) of the games could be guessed based on the minority (resp. majority) schema. However, whereas the difference between the Yes- vs No-set in the minority schema is significant it is not so in the majority schema. This suggests that, for humans, the level of difficulty of the two subsets is similar. The results on Group A (removing turns) shows that on average Yes-turns are more informative than No-turns. As expected, the last turn in the Yes-set is quite informative: with only the last turn (Group C), humans’ accuracy drops of only 10% (resp. 15.34%) reaching 88% (resp. 71.33%) accuracy in the minority (resp. majority) schema. Furthermore, the last turn in the Yes-set is quite redundant with the information provided by the previous turns: when receiving the dialogue without the last turns (Group B), humans’ accuracy drops of only 6% (resp. 11.34%) in the minority (resp. majority) schema. Instead, the last turn in the No-set seems to provide further information that needs to be integrated with those received in the previous turns:

without the last turn the accuracy on the No-set drops of 25.33% (resp. 31.34%). All in all, these results show that also for humans gathering information from the No-turn is harder than with the Yes-turn, yet the last turn in the No-set is informative and humans manage to profit from it to succeed in the task relatively well. This result highlights the value of negation in visual dialogues, and show why it is an important requirement for computational models to properly process it.

To measure the processing cost of negative turns, we have analyzed the average time taken by human to correctly solve games belonging to the four categories we discussed so far.

Table 5.6 shows that interpreting questions answered positively is faster than interpreting the ones answered negatively, and this result holds for all settings. In particular, processing positively-answered questions takes less than processing negatively-answered ones (group A), and a final positive turn is processed much faster than a negative final turn (group C). Interestingly, in the Yes-set guessing the target is faster when receiving the full dialogue than when receiving the dialogue without the last turn (0.72 vs. 0.94 seconds/token, p -value < 0.05), this might be due to what observed above, namely the last Yes-turn summarises the salient information collected till that point and hence speeds up the choice. Whereas the negative answer in the last turn brings a boost in performance, it does not affect significantly the time taken by human annotators to process the dialogue (0.79 vs. 0.85 seconds/token, p -value > 0.05). These results show that the time taken by human participants to solve the game mirrors the processing cost of negation, which is also influenced by the context (dialogue) in which it appears.

Results Humans vs. Models We now evaluate the models on the same controlled sample of games we used with human annotators. In Table 5.7, we report the task accuracy obtained by models when removing all the Yes turns (remaining with only No-turns) or all the No-turns (remaining with only Yes-turns). As can be seen from the table, the performance of the two best models is rather similar: both in the full dialogue and in the only Yes-turns the difference between their results is not significant. Similarly to humans, models accuracy drops less when receiving only the Yes-turns than when receiving only the No-turns. However, models' overall accuracy when receiving the full

		Full dialogue	Removing turns	
			Only No-turns	Only Yes-turns
BLIND	Random	16.5	16.5	16.5
	LSTM	57.0	30.67	48.00
	RoBERTa-S	54.66	29.33	50.00
	RoBERTa	60.0*	35.33	52.00**
MM	V-LSTM	55.66	25.33	50.66
	LXMERT-S	54.33	32.66	48.00
	LXMERT	59.67*	25.33	56.66**
Human (MAJ)		83.67	46.00	66.00

Table 5.7: **Controlled Sample.** Removing turns: comparison of the task accuracy when models receive the full dialogue vs. only the No- vs. only the Yes-turns. Human accuracy computed with the majority vote. (*, ** not significant)

		Yes-set			No-set		
		Full dialogue	W/o last	Only last	Full dialogue	W/o last	Only last
BLIND	LSTM	68.00	55.30	51.30	46.00	34.00	30.00
	RoBERTa-S	64.67	49.33	48.67	44.67	39.33	27.33
	RoBERTa	71.33	55.33	63.33	48.67	40.67	22.00
MM	V-LSTM	60.67	49.33	49.33	50.67	34.67	16.67
	LXMERT-S	61.33	50.00	47.33	47.33	36.00	22.00
	LXMERT	71.33	53.33	60.67	48.00	46.00	31.33
Humans (MAJ)		86.67	75.33	71.33	80.67	49.33	30.67

Table 5.8: **Controlled Sample.** Without the last turn, Only the last turn, Full Dialogue: accuracy comparison to highlight the role of the last turn when it contains a positive (Yes-set) vs negative (No-set) answer.

dialogue is far from the human upper-bound even when using the majority vote schema. As we can see in Table 5.8 this rather big difference between models and humans is due to the No-set: while humans correctly succeed in 80.67% of the games ending in a No-turn, models reach at most the 50%. It is thus clear that if it is true that negation has a higher processing cost for both humans and computational models, the latter struggle to profit from negatively answered questions.

5.6.2 Comparison with humans' errors

In the following, we run an error analysis by comparing models and humans on their failures. We expect that a model that properly grounds the dialogues is likely to make

	Removing turns	W/o last	Only last	Full dialogue
V-LSTM	80.65	73.56	78.61	53.38
LXMERT-S	82.12	71.35	81.12	59.12
LXMERT	83.05	77.48	85.19	58.68
RoBERTa-S	82.87	73.65	80.65	55.88
RoBERTa	83.43	72.44	82.56	55.00

Table 5.9: Error Analysis: Percentage of games human failed among those failed by each model.

human-like mistakes. To this end, among the games failed by a model, we check how many of them have been failed by at least one human annotator (Table 5.9); moreover, in the games in which a model and at least one participant failed, we check whether the error made by the model and the participant is exactly the same, i.e., if they have chosen the same (wrong) candidate object (Table 5.10). As we can see from Table 5.9, LXMERT is the model whose failed games are most similar to the ones failed by human annotators. However, if we look (in a more fine-grained way) at the exact candidate objects they select, we found that RoBERTa is the model whose errors are more human-like for most of the settings (Table 5.10). This analysis highlights how human annotations help interpret models’ results and evaluate the quality of their predictions.

In Figure 5.5, we report a game in which both models and humans failed to guess the target when the last turn was not given; interestingly, at that stage, with only the first three turns, the selection made by RoBERTa and humans could be valid. This shows that checking when models and humans make the same mistakes gives a hint about which errors are plausible. From our qualitative analysis, it seems that RoBERTa takes spatial questions into account more than LXMERT, maybe because it exploits the spatial coordinates of the candidate objects whereas LXMERT overrides that information with the one it receives from the visual features. More in-depth analysis is required to assess what factors most influence the outcome of the models.

5.6.3 Summary

The evaluation of models on the controlled sample confirms that RoBERTa and LXMERT behave rather similarly on the Yes-set across all settings. More interestingly, it shows

	Removing turns	W/o last	Only last	Full dialogue
V-LSTM	45.33	48.44	41.14	49.30
LXMERT-S	52.38	52.46	42.77	51.85
LXMERT	51.70	58.12	47.10	49.30
RoBERTa-S	57.33	51.22	46.67	44.74
RoBERTa	60.99	51.33	53.52	53.03

Table 5.10: Error Analysis: Percentage of games in which each model does the same mistake made by humans (i.e., by selecting the same wrong candidate object as a human annotator).

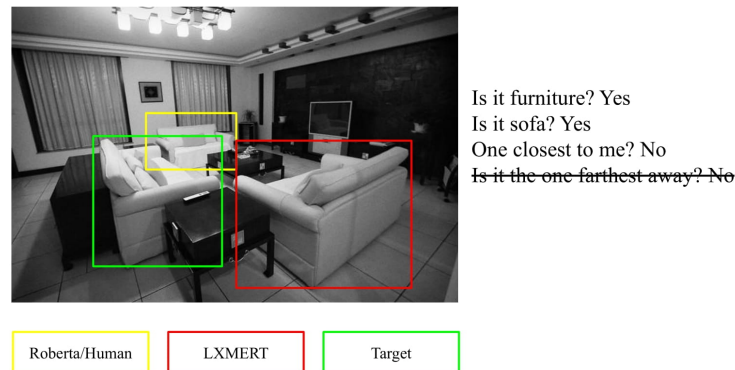


Figure 5.5: Errors made by humans and computational models when receiving dialogues without the last turn.

that in the No-set LXMERT is closer to humans than RoBERTa considering the accuracy in the task. LXMERT seems to be failing in the integration of the last No-turn with the dialogue history: its accuracy is similar to humans in the settings without last and only last turn, but it is far from them when the whole dialogue is given. Moreover, visual features seem to be of more help in the No-set than in the Yes-set: in the Yes-set across the controlled groups, the blind models do better or similar to their multimodal counterpart, whereas on the No-set the opposite holds. Finally, our error analysis reveals that RoBERTa is the model whose predictions are most human-like when it fails to identify the target object.

5.7 Discussion and Conclusion

In the current AI research, driven by the success of language models and neural network architectures, negation is under-studied. Dialogue history and visual context have been

shown to facilitate the processing of negation in humans. Hence, we took negation in visual dialogues as our object of investigation and studied how SOTA multimodal models profit from negatively answered questions (No-turns) in the GuessWhat?! game. Such negative information is informative for humans to succeed in the game, and this holds in particular when the No-turn occurs as the last one of a game in which the human player has been successful in guessing the target. Therefore, we focus attention on the subset of dialogues ending with a No-turn and compare them with those ending with a Yes-turn. Our results show that SOTA models' performances on these two sub-sets is rather different, eg., LXMERT obtains 71.9% vs. 50.9% accuracy in the Yes- vs. No-set, respectively (Table 5.2). To better interpret these results, we have run an online experiment with humans: we carefully selected a controlled sample of games and asked subjects to play the role of the guesser. We evaluated models' behaviour on such a controlled sample of games and used humans' results to better interpret the success and failures of models. The analysis shows that humans are much faster in processing positively answered questions than negatively answered ones. Yet, they do profit from the latter to succeed in the referential guessing task reaching 80.67% accuracy in the No-set – on which models guess correctly barely the 50% (Table 5.8). This shows that models are far away from the human ability to ground negation and we believe efforts should be put to reduce this important gap between humans and models' performance.

Our findings can help design models which could ground negation better than current SOTA models. First of all, our comparison between the accuracy obtained by LXMERT and RoBERTa in the various settings (Tables 5.2, 5.3 and 5.8) suggests that LXMERT grounding of negation within a dialogue could be improved by pre-training it on longer text. One could consider adding task-oriented dialogues in the pre-training phase. Moreover, our comparison of models' and humans' errors leads us to conjecture that LXMERT fails to exploit the spatial information provided in the dialogue, this could be behind the fact that though it grounds negation in short texts better than RoBERTa, the latter's mistakes are more human-like, since humans rely on such information to locate and identify the target object. This limitation of the LXMERT based Guesser could be overcome by building a model that exploits the image regions received as input to perform the task, similarly to what has been recently proposed in Tu et al. (2021) for

another multimodal model. Finally, (Hosseini et al., 2021) shows that pre-trained language models can better understand negation if trained with an unlikelihood objective. This is a first important step ahead in modelling negation in the neural network-era, but the model’s performance on entailment judgments involving negation is still low. Cognitive sciences findings on human processing of negation show that humans profit from expectations driven by the visual context to process negative information quickly and effectively (Nordmeyer and Frank, 2014); we believe that models should be trained to exploit more such expectations and that a (multimodal) communicative setting can help bring a boost for learning to encode (grounded) negation.

The results we obtain do not always provide conclusive answers, but we believe they convincingly show the weakness of current multimodal encoders in processing negation and represent a starting point towards future research. We started from the observation that dialogue history and visual context makes the processing of negation easier for humans. To fully understand whether this can be the case for models too, a comparison on processing negation in language-only vs multimodal settings should be carried out. To this end, the study could be extended to other datasets in which the visual input is not shared or only partially shared by the agents, such as VisDial, PhotoBook and Meet up! (Haber et al., 2019; Ilinykh et al., 2019a) or language-only task-oriented dialogues (Wu et al., 2020). Moreover, negative information can be conveyed in different ways, but we have studied only the easiest: a straightforward negative answer to a binary question. It would be interesting to explore the use of negation in declarative sentences and in more complex interactions. Finally, though our study builds on observations about the information gain the guesser accumulates through the dialogue at each turn, we have taken the dialogues as static blocks. A study about how humans and models incrementally gain information through the dialogue should be run to better understand their behaviour.

To conclude, our findings have theoretical and also practical implications: for humans, negatively answered questions can be as informative as affirmatively answered ones; a system that is not able to properly handle negation may be detrimental in real-world scenarios. More research should be done on the issue to better understand whether neural

network architectures can learn to ground negation on the alternative set it activates. To this end, we might need to single out various issues that are entangled in our analysis. First of all, it would be beneficial to have a multimodal dataset designed for this purpose. Secondly, when evaluating universal encoders the difference in the pre-training data is a confounder that should be avoided. Finally, it would be useful to have a large-scale human behavioural experiment that takes into account the incremental information gain at the core of a task-oriented dialogue exchange. We believe such data to be crucial both for training models to properly ground negation and for evaluating not only their task success but also their inside mechanisms as advocated for instance by [Zhang et al. \(2019\)](#). Once models learn to encode negation in grounded contexts, the next step will be to transfer such skills to language-only settings by exploiting transfer learning methods ([Ruder, 2019](#)).

5.8 Summary

So far, we discovered that:

- + Pre-Trained Transformers seem to understand the structure of a dialogue, identifying questions and answers;
- However, they struggle with negation, since they seem to identify the target object mostly thanks to positively answered questions;
- They perform much worse than humans, who are able to profit also from negatively answered questions.

Chapter 6

Pre-trained Transformers Integrating Complementary Multimodal Information

Current multimodal learning tasks often require them to combine *redundant* information provided by language and vision. Inspired by real-life communicative contexts, we propose a novel task where either modality is necessary but not sufficient to make a correct prediction. In this chapter, we first build a dataset of images and corresponding sentences provided by human participants. Second, we evaluate multimodal pre-trained Transformers on the benchmark and we compare their performance against other models and human speakers. We show that, while the task is relatively easy for humans, best-performing models such as pre-trained Transformers struggle to achieve similar results.

This chapter describes the work by (Pezzelle et al., 2020).

6.1 Introduction

Human communication, in real-life situations, is multimodal (Kress, 2010): To convey and understand a message uttered in natural language, people build on what is present in the multimodal context surrounding them. As such, speakers do not need to “repeat” something that is already provided by the environment; similarly, listeners leverage information from various modalities, such as vision, to interpret the linguistic message. Integrating information from multiple modalities is indeed crucial for attention and perception (Partan and Marler, 1999) since combined information from concurrent modalities can give rise to different messages (McGurk and MacDonald, 1976).

The argument that language and vision convey different, possibly complementary aspects of meaning has been largely made to motivate the need for multimodal semantic representations of words (Baroni, 2016; Beinborn et al., 2018). However, computational approaches to language and vision typically do not fully explore this complementarity. To illustrate, given an image (e.g., the one depicted in Figure 6.1), popular tasks involve describing it in natural language, e.g., “A tennis player about to hit the ball” (Image Captioning; see Bernardi et al., 2016b); answering questions that are grounded in it, e.g., Q: “What sport is he playing?”, A: “Tennis” (Visual Question Answering; see Antol et al., 2015a); having a dialogue on its entities, e.g., Q: “Is the person holding a racket?”, A: “Yes.” (visually-grounded dialogue; see De Vries et al., 2017; Das et al., 2017c). While all these tasks challenge models to perform visual grounding, i.e., an effective *alignment* of language and vision, none of them require a genuine *combination* of complementary information provided by the two modalities. All the information is fully available in the visual scene, and language is used to describe or retrieve it.

In this work, we propose a novel benchmark, *Be Different to Be Better* (in short, **BD2BB**), where the *different*, complementary information provided by the two modalities should push models develop a *better*, richer multimodal representation. As illustrated in Figure 6.1, models are asked to choose, among a set of **candidate actions**, the one a person who sees the visual context depicted by the **image** would do based on a certain **intention** (i.e., their goal, attitude or feeling). Crucially, the resulting multimodal input

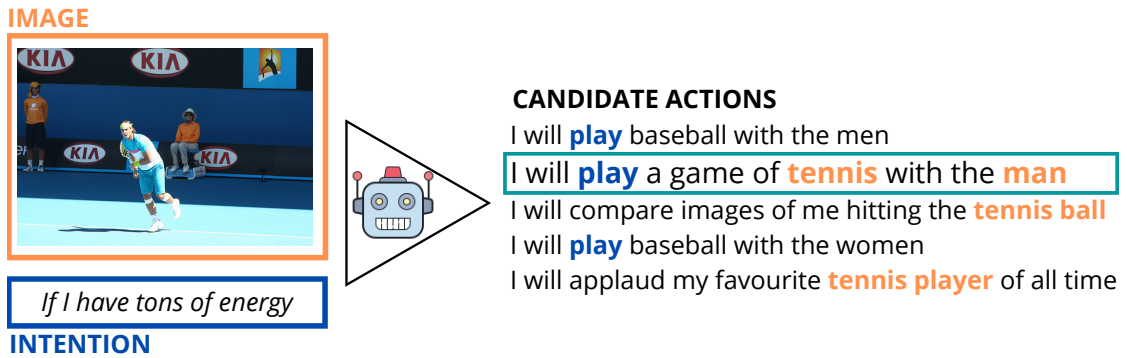


Figure 6.1: One real sample of our proposed task. Given an **image** depicting, e.g., a tennis player during a match and the **intention** “*If I have tons of energy*”, the task involves choosing, from a list of 5 **candidate actions**, the target action that unequivocally applies to the combined multimodal input: “*I will play a game of tennis with the man*”. The task is challenging: a model exploiting a language or vision bias could fall into the trap of decoy actions containing words highlighted in blue or orange, respectively. Therefore, selecting the *target* action requires models perform a genuine integration of the two modalities, whose information is complementary. Best viewed in color.

(the sum of the image and the intention) will be richer compared to that conveyed by either modality in isolation; in fact, the two modalities convey complementary or *non-redundant* information (Partan and Marler, 1999).

To illustrate, a model that only relies on the (non-grounded) linguistic information conveyed by the intention, i.e., “*If I have tons of energy*”, might consider as equally plausible any actions that have to do with playing a sport, e.g., “*I will play baseball with the men*” or “*I will play a game of tennis with the man*”. Similarly, a model that only relies on the visual information conveyed by the image—a tennis player during a match—might consider as equally plausible any actions that have to do with ‘tennis’ and/or ‘player’, e.g., “*I will applaud my favourite tennis player of all time*” or “*I will play a game of tennis with the man*”. In contrast, a model that genuinely combines information conveyed by both modalities should be able to select the *target* action, namely the only one that is both consistent with the intention and grounded in the image, i.e., “*I will play a game of tennis with the man*”. Moreover, similarly to real-life communicative scenarios, in our approach different language inputs *modulate* differently the same visual context, and this gives rise to various multimodal messages. To illustrate, if the image in Figure 6.1 is paired with the intention “*If I am tired watching*”, the target action “*I will play a game of tennis with the man*” is no longer valid. Indeed, the target

action in this context is “*I will leave the tennis court*” (see Figure 6.3).

Our work has the following key contributions:

- We introduce a novel multimodal benchmark: the set of $\sim 10\text{K}$ $\langle image, intention, action \rangle$ datapoints collected via crowdsourcing and enriched with meta-annotation; the multiple choice task, **BD2BB**, which requires proper integration of language and vision and is specifically aimed at testing SoA pretrained multimodal models. The benchmark, together with the code and trained models, is available at: <https://sites.google.com/view/bd2bb>;
- We test various models (including the SoA multimodal, transformer-based LXMERT; Tan and Bansal, 2019) and show that, while **BD2BB** is a relatively easy task for humans ($\sim 80\%$ acc.), best systems struggle to achieve a similar performance ($\sim 60\%$ acc.);
- We extensively analyze the results and show the advantage of exploiting multimodal pretrained representations. This confirms they are effective, but not enough to solve the task.

6.2 Related Work

Since the introduction of the earliest multimodal tasks, such as Image Captioning (IC; see Bernardi et al., 2016b) and Visual Question Answering (VQA; Antol et al., 2015a), a plethora of tasks dealing with language and vision have been proposed. In parallel, baseline models have been replaced by more powerful attention-based systems (Anderson et al., 2018b) and, more recently, by transformer-based architectures pretrained on several tasks (Tan and Bansal, 2019; Lu et al., 2019b; Chen et al., 2019b). These latter models build on multimodal representations that are meant to be task-agnostic; as such, they can be transferred to virtually any other multimodal task with minimal fine-tuning. Our work contribute to these two lines of research by (1) introducing a novel multimodal task, and (2) by evaluating a SoA multimodal encoder on it.

Multimodal tasks VQA was originally proposed to overcome the challenge of quantitatively evaluate IC models. Driven by VQA, several datasets have been proposed to minimize the bias observed in natural images (Goyal et al., 2017; Ray et al., 2019); to force models to “reason” over a joint visual and linguistic input (Johnson et al., 2017b; Suhr et al., 2019); to deal with objects’ attributes and relations (Krishna et al., 2017b); to encompass more diverse (Zhu et al., 2016) and goal-oriented questions and answers (Gurari et al., 2018b). At the same time, some work proposed higher-level evaluations of VQA models and showed their limitations (Hodosh and Hockenmaier, 2016; Shekhar et al., 2017b); similarly, recent attention has been paid to understand what makes a question “difficult” for a model (Bhattacharya et al., 2019; Terao et al., 2020). Despite impressive progress, current approaches to VQA do not tackle one crucial limitation of the task: the answer to a question is given by the *alignment* of language and vision rather than their *complementary* integration.

Moving from objects to *actions*, several tasks have been proposed to mimic more realistic settings where a higher degree of integration between modalities is required. One is visual storytelling (Huang et al., 2016; Gonzalez-Rico and Pineda, 2018; Lukin et al., 2018), where models have to understand the action depicted in each photo and their relations to generate a story. Similar abilities are required in the task of generating non-grounded, human-like questions about an image (Mostafazadeh et al., 2016; Jain et al., 2017), and in that of asking discriminative questions over pairs of similar scenes (Li et al., 2017). Related tasks are also those of predicting motivations of visually-grounded actions (Vondrick et al., 2016) or generating explanations for a given answer (Park et al., 2018; Hendricks et al., 2018).

An even higher level of understanding of vision and language is required in the tasks of filling the blank with the correct answer (Yu et al., 2015); answering questions from videos and subtitles (Lei et al., 2018); having a dialogue on objects (De Vries et al., 2017; Das et al., 2017c) or events (Mostafazadeh et al., 2017); answering and justifying commonsense questions (Zellers et al., 2019). However, all these tasks require making *commonsense* inferences over the two modalities rather than integrating their complementary information to answer a *grounded* question.

More akin to ours are the approaches by [Iyyer et al. \(2017\)](#), which aims to predict the subsequent scene and dialogue in a comic strip, and [Kruk et al. \(2019\)](#), where the goal is to compute the communicative intent of a social media post. Though they both require a challenging integration of language and vision, these tasks (as well as the type of data they use) are crucially different from **BD2BB**, where the task is to predict the action that is consequent to a given intention based on the image.

Transformer-based multimodal models Developing universal multimodal encoders whose pretrained representations are suitable for virtually any multimodal task is a crucial challenge. Inspired by the success of BERT, several pre-trained Transformers have been recently proposed in the domain of language and vision ([Lu et al., 2019b](#); [Tan and Bansal, 2019](#); [Chen et al., 2019b](#); [Su et al., 2020](#); and [Nan Duan et al., 2020](#)). While these architectures achieve state-of-the-art performance in many tasks, their novelty and complexity leave several questions open, and further work is needed to better understand, e.g., which layers are more suitable for transferability ([Tamkin et al., 2020](#)), or what is the relation between pretraining and downstream tasks ([Zamir et al., 2018](#); [Singh et al., 2020](#)). Moreover, to prove they are readily applicable to novel multimodal benchmarks, pretrained universal encoders should be ideally effective with only minimal fine-tuning on the target tasks.

In this light, we believe that more efforts should be put in developing datasets that are challenging and yet relatively small, in line with the ‘diagnostic’ datasets proposed for VQA ([Johnson et al., 2017b](#)) and the easy vs. hard subsets introduced by [Akula et al. \(2020\)](#) for visual referring expression recognition. Our contribution follows this line of thought.

6.3 Data

In this section, we describe how we collected intentions and actions through crowdsourcing, and the subsequent phase of data meta-annotation. Consistently with our purposes, we needed images that elicit goals and feelings (the intentions) in the annota-

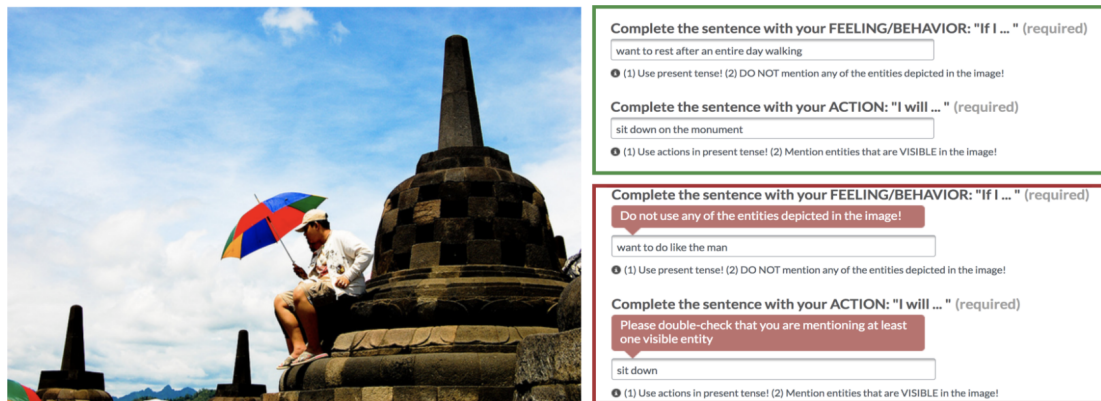


Figure 6.2: Data collection. Examples of good (top) and bad (bottom) annotations provided to participants in the task instructions. Errors and corresponding warnings are shown to make participants familiarize with the tool.

tors, as well as consequent actions. To this end, we used the partition of the MS-COCO dataset (Lin et al., 2014b) provided by Vondrick et al. (2016),¹ where each of the 10, 191 images depicts at least one person. This choice was aimed to make the participants' task more natural: indeed, the presence of people in the image allows more possibilities of interaction, and therefore guarantees that some actions can be performed in that situation.

6.3.1 Data Collection

We set up an annotation tool on Figure-Eight² (see Figure 6.2) where annotators were shown an image and asked to imagine themselves being in that situation, as ideal observers not represented in the picture. We instructed them to carefully look at the image and think about 1) an intention, i.e., *how they might feel/behave if they were in that situation*; 2) an action, i.e., *what they would do based on that feeling/behavior*. Intentions and actions were typed in free form by participants in two separate text boxes; by instructions, their sentences had to complete the provided opening words *If I...* and *I will...*, respectively. To ensure that intentions conveyed information that was complementary (non-redundant) to that by the image, participants were instructed not to

¹http://visiondl.cs.umbc.edu/webpage/codedata/intention/motivations_clean.zip

²<https://www.figure-eight.com/>

INTENTIONS	ACTIONS
1. <i>If I want to be on a spotlight</i>	→ 1. <i>I will stay behind the player</i>
2. <i>If I want to give encouragement</i>	→ 2. <i>I will applaud the player</i>
3. <i>If I want to make my dream come true</i>	→ 3. <i>I will have to win the tennis match</i>
4. <i>If I have tons of energy</i>	→ 4. <i>I will play a game of tennis with the man</i>
5. <i>If I get tired of watching</i>	→ 5. <i>I will leave the tennis court</i>

Figure 6.3: Five $\langle intention, action \rangle$ tuples provided by 5 unique participants for the image in Figure 6.1.

mention *any* of the entities (people, objects, etc.) shown in the image. In contrast, to ensure that actions contained information that was grounded in the image, participants were asked to mention at least one visible entity when writing their action (see errors and warnings in Figure 6.2).³

We randomly selected $\sim 3.6\text{K}$ images from the split by Vondrick et al. (2016) and, for each of them, we collected on average 5 $\langle intention, action \rangle$ tuples by 5 participants. In total, $\sim 18\text{K}$ unique $\langle image, intention, action \rangle$ datapoints were collected. Participants were recruited from native-English countries only. Overall, 477 annotators (based on the IP) took part in the data collection; on average, each of them provided 38 annotations. Participants were paid 0.04\$ per tuple.⁴ In total, the data collection costed $\sim 900\text{\$}$.

A few filtering steps were needed to get rid of datapoints with invalid annotations. First, we discarded those datapoints where intentions and/or actions were either not in English (e.g., bot-generated *Lorem Ipsum* sequences) or nonsense strings (e.g., random sequences of characters). This step was done semi-manually and filtered out $\sim 3\text{K}$ datapoints. Second, we removed datapoints where the action did not contain any noun nor pronoun. After this, we were left with 12,457 valid datapoints.

To illustrate the type of data collected, Figure 6.3 reports the 5 $\langle intention, action \rangle$ tuples provided by 5 annotators for the image in Figure 6.1. As can be noted, the same visual context elicits different intentions, which in turn give rise to different possible actions. Crucially, no intentions refer to anything that is visible in the image, which

³Further details on data collection and meta-annotation, dataset and models are given in the appendix

⁴This corresponds to a hourly wage of around 8\$/hour.

makes them suitable for virtually any visual context. As for the actions, in contrast, they all 1) mention at least one entity that is *grounded* in the given scene, e.g., “player” or “tennis court”, which makes them plausible only for sports contexts, particularly ‘tennis’; 2) match their corresponding intention, but not (or to a much lesser extent) the others; i.e., different intentions trigger different actions, and the verb in the action is a proxy for such diversity. Below, we describe the meta-annotation process we performed to categorize each datapoint with respect to: 1) the topic of its action, e.g., ‘tennis’; and 2) the argument structure of the verbs in its action.

6.3.2 Meta-Annotation

Topic For each of the 12,457 datapoints, we built a 512-d semantic representation of its action using the off-the-shelf Universal Sentence Encoder (USE; Cer et al., 2018). We then run a k -means clustering algorithm over these vectors and obtained 60 *topic* clusters.⁵ By manual inspection, 54 clusters were found to consistently group together actions revolving on the same topic, e.g., ‘tennis’ or ‘birthday’, in a way that it was easy to label them using such terms. Since for the remaining 6 clusters this was not straightforward due to the presence of rather disconnected actions, we filtered these clusters out. We further polished the 54 clusters (a) by manually moving actions to clusters that fit them better, and (b) by removing actions that were not in line with the cluster topic. Moreover, we removed actions that did not comply with the instructions provided to annotators during the data collection. After these steps, we were left with 10,287 $\langle image, intention, action \rangle$ datapoints.

Argument structure Using the Stanford NLP Parser (Chen and Manning, 2014), we annotated the actions in each of the 10,287 topic-categorized datapoints by means of a 4-code annotation schema. In particular, from each parsed action we extracted its main verb (code1) and its direct or indirect object (code2). Moreover, when present, the verb of the coordinate or subordinated sentence was also extracted (code3), as well as other

⁵The best number of clusters was chosen based on the Elbow method, which relies on cluster consistency.

nouns in any complement position of the main or secondary verb (code4).⁶ All the outputs by the parser were manually checked and fixed where needed. Given the action “*I will swing the racket to hit the ball*”, for example, we thus obtained the following argument structure annotation: $\langle swing \rangle$ (code1), $\langle racket \rangle$ (code2), $\langle hit \rangle$ (code3), $\langle ball \rangle$ (code4). As can be seen, this simplified representation of the action provides information on both its verbs (that are *consequent* to the intention) and nouns (*grounded* in the image). The 10,287 annotated datapoints were used to build the dataset for our task.

6.4 Task

We introduce the *Be Different to Be Better* (**BD2BB**) task, where the *different*, i.e., complementary information provided by the two modalities should push models develop a *better*, i.e., richer multimodal representation. To evaluate these abilities, we frame our task as a multiple-choice problem (similar qto Antol et al., 2015a; Yu et al., 2015; Zhu et al., 2016) where either modality is necessary but not sufficient to perform a correct prediction. The task is the following (see Figure 6.1): given an image and a corresponding intention, the model has to choose the correct action over a set of 5 candidate actions. We refer to the correct action as the *target* action; to the wrong actions as the *decoy* actions. Similarly to Chao et al. (2018), decoy actions are carefully selected to be as plausible as possible when evaluated against either the intention (2 decoys) or the image (the other 2) only. Below, we explain how language-based and image-based decoys were selected based on the meta-annotation.

Language-based decoys For each of the 10,287 $\langle image, intention, action \rangle$ datapoints, we randomly selected a number of datapoints from the entire data that had the following criteria: 1) their action belonged to a different topic cluster than the one including the target action; 2) their action did not share any noun with the target action, i.e., their $\langle code2 \rangle$ and $\langle code4 \rangle$ were different. We then computed a similarity score between the target action and each of these selected actions by means of the cosine of their

⁶While verbs were lemmatized, we did not do so for nouns due to the visual difference between, e.g., *player* and *players*.

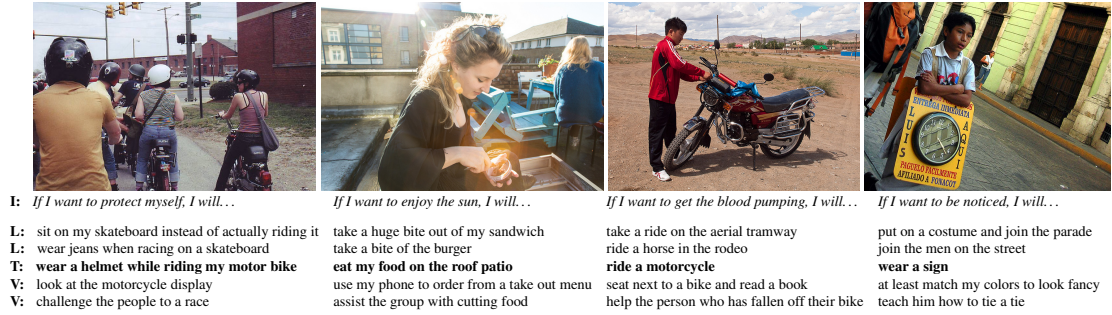


Figure 6.4: Four samples from our dataset. **I:** Intention; **T:** Target action; **L/V:** Language-/Vision-based decoys.

	#samples (%)	#img	#int	#act	#t-act	#d-act	avg int len	avg act len
train	2102 (20%)	1517	1683	5063	2102	4228	22.15	35.34
val	4082 (40%)	2447	2772	6082	3567	4133	20.76	36.20
test	4081 (40%)	2425	2720	6108	3561	4138	20.49	36.00
total	10265 (100%)	3215	6192	8751	8738	6339	20.94	35.94

Table 6.1: Descriptive statistics of the dataset including, from left to right: 1) # (and %) of unique samples; 2) # of unique images; 3) # of unique intentions; 4) # of unique actions; 5) # of unique target actions; 6) # of unique decoy actions; 7) average number of tokens in intentions; 8) average number of tokens in actions.

USE representations. We ranked these scores and selected as our language decoys the two with the highest similarity. This way, we obtained language-based decoys that are semantically very similar to the target action, but are on a different topic and do not share any noun with it.

Vision-based decoys For each datapoint, we randomly selected a number of datapoints from the entire data that had the following criteria: 1) their action belonged to the same topic cluster of the target one; 2) their action did not share any verb with the target action, i.e., their $\langle code1 \rangle$ and $\langle code3 \rangle$ were different. We then ranked these actions with respect to their USE similarity with the target one, and selected as our vision-based decoys the two with the lowest score. This way, we obtained vision-based decoys that are about the same topic of the target action; at the same time, they do not share any verbs with it and are semantically different.

6.4.1 Dataset

Our final dataset includes 10,265 samples⁷ as the ones depicted in Figure 6.4: each sample consists of a unique $\langle image, intention, action \rangle$ datapoint paired with 4 carefully-selected decoy actions. Consistently with our purpose of making **BD2BB** a challenging benchmark for pretrained multimodal architectures (see Section 6.1), we split the dataset into “unusual” train/val/test partitions; i.e., we selected 20% samples for training; the remaining for validation (40%) and test (40%). We propose having small training data and larger validation and test sets should become a standard, as pretrained models already build on a massive amount of data.

Table 6.1 reports the descriptive statistics of the dataset, including the number of unique images, intentions and actions per split, and the average length of the sentences. All the experiments reported in the chapter are performed on these splits.

6.5 Experiments

To test the importance of combining information from the two modalities and the independent contribution of either modality, we experiment with 3 settings of the **BD2BB** task: L , where the target action among the 5 candidates has to be guessed based on the intention only; V , where only the image is provided; LV , where both the image and the intention are provided. For each setting of the task, we evaluate the performance of (1) a simple baseline trained from scratch on the task; (2) a state-of-art transformer-based pretrained model fine-tuned on the task; (3) the same transformer-based model trained from scratch on the task. Moreover, results by models are compared to (4) human performance.

⁷For 22 datapoints it was not possible to find all the decoys, hence they were discarded during the creation of the dataset.

6.5.1 Models

Baseline For each $\langle image, intention, action \rangle$ datapoint in the sample, $baseline_{LV}$ builds a multimodal representation by concatenating the 2048-d visual features of the image (extracted from a pretrained ResNet-101; ?) with the 300-d embedding of the intention and the 300-d embedding of the action. Embeddings for both the intention and the action are obtained by summing the GloVe embeddings (Pennington et al., 2014) of the words in them. The concatenated features are linearly projected into a vector (8192-d), passed through ReLU, and linearly projected into a single value. Softmax probabilities are computed over the 5 sample’s candidate values. The $baseline_L$ only concatenates intention and action embeddings (600-d representation); $baseline_V$ concatenates the visual features with the action embedding (2348-d). Finally, to account for any bias due to unavoidable association and repetition patterns among the actions, we test a version of the baseline which only encodes the actions. We refer to it as *actions-only*.

RoBERTa In setting L , we employ RoBERTa (Liu et al., 2019a), which achieves best-performing performance in the challenging multiple-choice SWAG task (Zellers et al., 2018). We adapt $RoBERTa_{BASE}$ to our task as following: for each of the 5 $\langle image, intention, action \rangle$ datapoints in the sample, RoBERTa encodes the input as a sequence composed by $\langle CLS \rangle$, the intention, $\langle SEP \rangle$, the action, and $\langle EOS \rangle$. The encoding corresponding to the $\langle CLS \rangle$ token (768-d) is passed through Tanh, linearly projected into a vector (768-d), passed to Dropout (Srivastava et al., 2014), and linearly projected into a single value. Softmax probabilities are computed over the 5 sample’s candidate values. As mentioned above, we evaluate two model versions: $RoBERTa_L$, pretrained and fine-tuned on our task, and $RoBERTa_L^s$, trained from scratch on BD2BB.

LXMERT In settings LV and V , we employ LXMERT (Tan and Bansal, 2019) Like RoBERTa, LXMERT uses the special tokens $\langle CLS \rangle$ and $\langle SEP \rangle$ but, differently from RoBERTa, here $\langle SEP \rangle$ is used both to separate sequences and to denote the end of the textual input. Hence, we take this into account when adapting LXMERT to our task.

Similar to RoBERTa, we use the encoding corresponding to $\langle CLS \rangle$ (768-d) to obtain a probability distribution over the 5 sample’s candidate values. For each task setting, we evaluate each model in two versions, i.e., pretrained model fine-tuned on our task ($LXMERT_{LV}$ and $LXMERT_V$); trained from scratch ($LXMERT_{LV}^s$ and $LXMERT_V^s$).

Experimental setup For baseline models, we perform hyperparameter search on learning rate, Dropout, and hidden size; as for transformer-based models, we use the best configurations reported in the source papers (reproducibility details in the appendix). All models are trained with 3 random seeds for 50 epochs with Adam (Kingma and Ba, 2015) minimizing a Cross Entropy Loss between the probability distribution over the 5 sample’s candidate actions and the ground-truth action. For each of the 3 runs, we consider the model with the highest validation accuracy. Average accuracy and standard deviation over 3 runs is computed.

6.5.2 Human Evaluation

We randomly extracted 300 unique samples from the dataset and split them into 3 partitions including 100 samples each. For each partition, we collected judgments by 3 participants in each setting of the task: L , V , and LV . Crucially, participants did the task only once per partition; i.e., they judged each sample only in one of the 3 task settings. Using Quiz Maker,⁸ we collected 2,700 unique responses from 11 subjects who participated on a voluntary basis. For each setting of the task, we counted as ‘correctly predicted’ the samples where at least 2 out of 3 annotators converged on the target action. Moreover, for each task setting we computed the ‘best’ accuracy, i.e., the average of the 3 participants who achieved the highest accuracy in each split.

⁸<https://www.quiz-maker.com>

		model	accuracy	
			val \pm std	test \pm std
SCRATCH		actions-only	44.0 \pm 0.4	44.6 \pm 0.8
		baseline _L	45.3 \pm 0.9	45.9 \pm 0.9
		baseline _V	45.8 \pm 0.8	46.1 \pm 0.8
		baseline _{LV}	48.6 \pm 0.9	49.0 \pm 0.9
SCRATCH		RoBERTa _L ^s	47.0 \pm 0.2	47.2 \pm 0.1
		LXMERT _V ^s	30.9 \pm 0.9	31.8 \pm 0.4
		LXMERT _{LV} ^s	50.4 \pm 0.3	51.3 \pm 0.4
PRETRAIN		RoBERTa _L	55.9 \pm 0.9	56.2 \pm 1.3
		LXMERT _V	59.1 \pm 0.2	59.2 \pm 0.6
		LXMERT _{LV}	62.8 \pm 2.3	62.2 \pm 2.2
		humans _L	50.0 (best 54.0)	
		humans _V	72.3 (best 73.7)	
		humans _{LV}	79.0 (best 82.3)	
		<i>chance</i>	20.0	20.0

Table 6.2: Results for the 3 settings: L , V , and LV . ^s refers to transformer-based models trained from scratch. For each model, we report average accuracy and std over 3 runs. Human accuracy is computed over 300 samples (we report values based on both majority vote, i.e., 2 out of 3, and average of best participants; see 6.5.2).

6.6 Results

Results by both models and humans are reported in Table 6.2. Several key observations can be made.

Multimodal integration is the key. The overall best-performing model in **BD2BB** is LXMERT_{LV} (62.2%), which outperforms the other pretrained models, i.e., RoBERTa_L (56.2%) and LXMERT_V (59.2%). On the one hand, this shows that disposing of both modalities is beneficial to perform the task. This is in line with the results by human participants, who achieve the highest accuracy in the multimodal setting (79% vs. 50% of L and 72.3% of V). On the other hand, the finding that LXMERT_V surpasses RoBERTa_L (+3%) confirms that the image provides more information compared to the intention. This, again, is consistent with human results, where the gap between V and LV (−7%) is much smaller compared to that between L and LV (−29%). For humans, this visual advantage is likely due to (MS-COCO) images depicting complex events that elicit a broad range of aspects related to people’s experience of the world. As for the models,

it confirms that LXMERT, thanks to its massive pretraining, is effective in extracting fine-grained information from images.

Models are far from humans. Humans achieve around 80% accuracy (‘best’ 82%) on the multimodal version of the task. This is a high result, in line with previous work with a similar setup (consider, e.g., SWAG, where ‘expert’ human accuracy is around 85% with 4 choices, i.e., chance level at 25%; Zellers et al., 2018). At the same time, the non-perfect human accuracy reveals that the benchmark is challenging due to the careful selection of plausible decoys. Compared to humans, the best-performing LXMERT_{LV} achieves much lower results (−17%), which indicates that **BD2BB** is challenging and far from being solved. Since the gap between best-performing models and human participants in unimodal settings is smaller (−13% in *V* and −6% in *L*), the biggest computational challenge lies in the integration of complementary information from different modalities.

Pretrained is better. Pretrained models neatly outperform the baseline in all the versions of the task⁹ and, more interestingly, also all their counterparts trained from scratch. As can be seen in Table 6.2, indeed, transformer-based models trained from scratch achieve results that are only slightly better than those by the baseline in both *LV* and *L*; as for *V*, LXMERT_V^s turns out to perform worse than the baseline_V^s (and even worse than the actions-only baseline). This clearly shows that these architectures are very effective when building on their pretraining, but suffer when challenged to learn a task from scratch with relatively few samples.

6.7 Analysis

Best models’ errors We perform an analysis on the errors made by the 3 pretrained models to check whether they fall more often into the language-based or vision-based

⁹It should be noted that the baselines are only slightly better than *actions-only*; this suggests that these models are only marginally capable of extracting (and combining) relevant information for the task from the image and the intention.



I: *If I am in the mood to act silly, I will...*

- L:** attend a dinner like this man holding a gift
- L:** buy him a cake and invite his friends to party
- T:** **act silly with this man and eat cake**
- V:** help my child cut their cake
- V:** have cake with soldiers



If I don't like this, I will...

- sit next to the woman on the bench
- get my face painted
- avert my eyes from the man who looks silly**
- teach him how to tie a tie
- wear a costume and march in a parade**

Figure 6.5: Two samples where humans give the correct answer in the LV setting—but neither in L nor in V . $LXMERT_{LV}$ picks the correct answer (**blue**) in the left sample, a wrong one (**red**) in the right sample. **I:** Intention; **T:** Target action; **L/V:** Language-/Vision-based decoys. Best viewed in color.

decoys. To do so, we focus on each model’s best run, and compute the proportion of wrong predictions in the test set that belong to one or the other decoy type. For comparison, a model that makes modality-balanced wrong predictions should fall into language-/vision-based decoys 50% of the times. Quite surprisingly, $RoBERTa_L$ has only a moderate bias toward language-based decoys: in fact, only 60.2% of its errors are of this type. As for $LXMERT_V$, no bias at all is observed toward the vision-based decoys (48.6%). Finally, the best-performing $LXMERT_{LV}$ is shown to be halfway between these models, with only a slight preference for language-based (55.1%) over vision-based decoys (44.9%).

In Figure 6.5, we report two cherrypicked examples where $LXMERT_{LV}$ either correctly predicts the target action (left) or chooses a wrong one, in this case a vision-based decoy (right). It is worth mentioning that these two cases are challenging: for both of them, human annotators were able to pick the correct action only in the multimodal version of the task—but neither in L nor in V . As can be seen, in the leftmost example the model

model	accuracy	humans
	<i>hard test \pm std</i>	
RoBERTa _L	55.1 \pm 1.6	56.5
LXMERT _V	56.9 \pm 0.8	73.9
LXMERT _{LV}	58.3 \pm 2.7	78.3

Table 6.3: Accuracy of the pretrained transformer-based models on the *hard* samples of the test set. Human accuracy is computed over 92 samples.

does a good job in combining complementary information from language and vision. In the righthmost one, instead, it picks an action that is very much plausible based on the image, but not in presence of the given intention containing a negation (*don't*). Taken together, these analyses indicate that no simple strategies can be exploited by models to detect and rule out decoy types. Language- and vision-based decoys are equally challenging, and combining complementary information is needed to solve the task.

Hard test To explore the robustness of the pretrained models, we check how well they perform on a subset of the test set where several features of the samples were *unseen* in training. In particular, neither the image nor the intention were seen in training; moreover, the target action could be seen as a decoy but never as the target. In Table 6.3 we report the results by the 3 pretrained models on this subset (1,505 samples); we refer to it as the *hard* test. As can be seen, all models experience a small decrease in accuracy compared to the whole test set—while humans do not. This indicates that the hard test is indeed more challenging. However, pretrained models are overall robust to unseen features. In line with the standard test set, LXMERT_{LV} still outperforms the unimodal models, though its drop in performance (-4%) is more pronounced compared to them ($-1/2\%$). This suggests that part of the advantage of the multimodal system over the unimodal ones is due to its fine-tuning. Indeed, pretraining on its own is not enough to properly combine complementary information from the intention and the image. Finally, since humans do not perform worse in these samples, the performance gap with LXMERT_{LV} increases to $\sim 20\%$.

6.8 Conclusion

Inspired by real-life communicative contexts where language and vision are *non-redundant*, we proposed a novel benchmark to challenge models to combine complementary multimodal information. This is a crucial ability that, we believe, our benchmark will contribute push further. In particular, recently proposed universal multimodal encoders can greatly benefit from relatively small but challenging resources as is **BD2BB**, which can be used to shed light on model abilities and help developing architectures which exhibit more human-like skills.

Here, we evaluated LXMERT and showed that it struggles to achieve results that are comparable to those by humans. In the future, we plan to evaluate other multimodal encoders on it, and to contribute to the development of better multimodal systems.

6.9 Summary

So far, we discovered that:

- + Pre-Trained Transformers can, to some extent, integrate complementary multimodal information;
- + Pre-trained models are much better than models trained from scratch;
- However, their accuracy is still far from human performance.

Chapter 7

Impact of Task Difficulty on Transfer Learning of Neural Multimodal Models

In this chapter, we study the impact of task difficulty on the ability of multimodal models to transfer previously-acquired knowledge to new tasks and to not forget what they have previously learned in VQA. In particular, motivated by evidence from psycholinguistics, we devise a set of linguistically-informed VQA tasks which differ by the types of questions involved (Wh-questions and polar questions). Then, we assess whether the order in which a child acquires question types facilitates transfer learning and mitigates forgetting. Our results show that training models on easier tasks first improves accuracy on the following tasks and mitigates forgetting of the previously-learned ones.

7.1 Introduction

Neural models are incapable of continuously learning new tasks, as they forget how to perform the previously-learned ones. This problem, called *catastrophic forgetting*, is prominent in ANNs (McClelland et al., 1995). *Continual Learning* (CL) addresses this

This chapter describes the work by Greco et al. (2019).

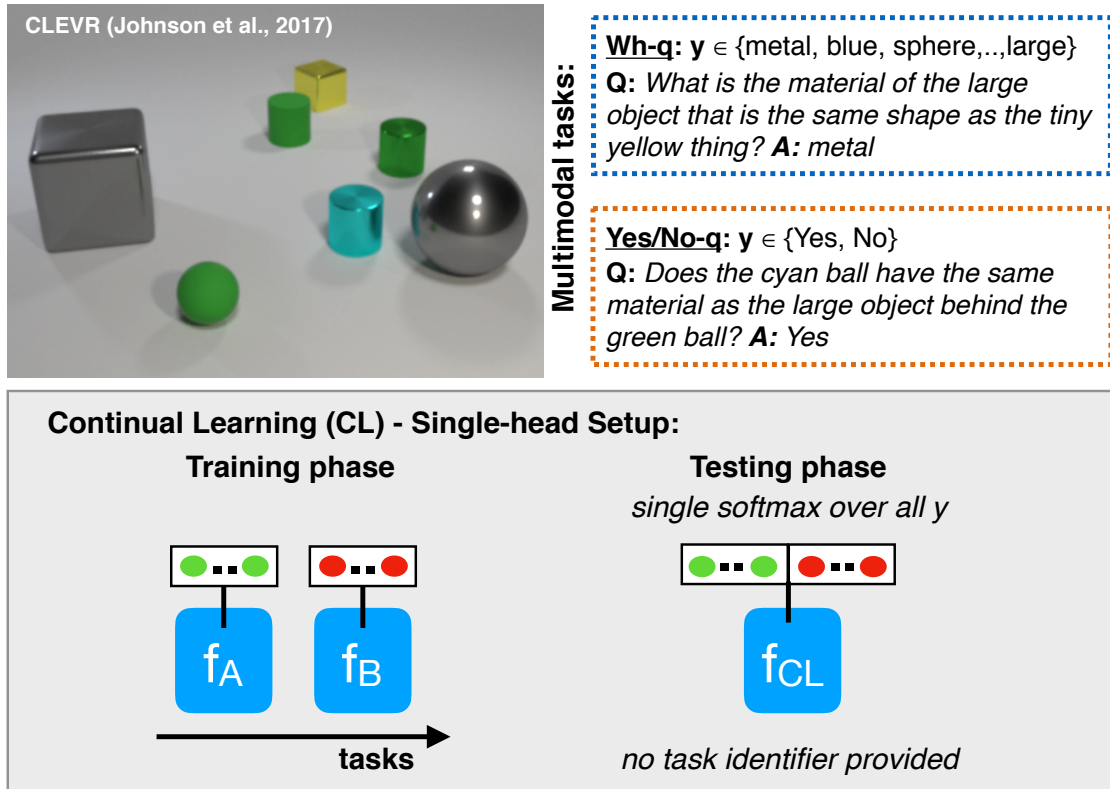


Figure 7.1: Overview of our linguistically-informed CL setup for VQA.

problem by trying to equip models with the capability to continuously learn new tasks over time in order to build models which are able to transfer previous knowledge to new tasks without forgetting what they have previously learned (Ring, 1997). CL has received considerable attention in Computer Vision (e.g., Zenke et al., 2017; Kirkpatrick et al., 2017), but far less attention within Natural Language Processing.

We investigate CL in the context of multimodal models for VQA (Antol et al., 2015b) motivated by evidence from psycholinguistics. Evidence from child language acquisition indicates that children learn Wh-questions before polar (Yes/No) questions (Moradlou and Ginzburg, 2016; Moradlou et al., 2018). Motivated by this finding, we design a set of linguistically-informed experiments: i) to investigate whether the order in which children acquire question types facilitates transfer learning and mitigates catastrophic forgetting of neural models; ii) to measure how far two well-known CL approaches help to overcome the problem (Robins, 1995; Kirkpatrick et al., 2017)

Contributions: Our study contributes to the literature on CL in Natural Language Processing. In particular: i) we introduce a CL setup based on linguistically-informed task pairs which differ with respect to question types and level of difficulty; ii) we show the importance of task order, an often overlooked aspect, and observe asymmetric synergetic effects; iii) our results show that training models on easier tasks first improves accuracy on the following tasks and mitigates forgetting of the previously-learned ones; iv) moreover, our VQA model suffers from extreme forgetting; rehearsal gives better results than a regularization-based method. Our error analysis shows that the latter approach encounters problems even in discerning Task A after having been trained on Task B. Our study opens the door to deeper investigations of CL on linguistic skills with different levels of difficulty based of psycholinguistics findings.

7.2 Task Setup

As a first step towards understanding the connection between linguistic skills and the impact on CL, we design a set of experiments within VQA where tasks differ with respect to the *type of question* and the *level of difficulty* according to the psycholinguistics literature. The overall setup is illustrated in Figure 7.1 and described next.

Dataset We rely on CLEVR, a diagnostic VQA dataset previously described in section 2.5.1 in order to build our multimodal tasks. In particular, we select CLEVR since it provides different question types testing several abilities of multimodal models.

Multimodal Tasks We select the CLEVR sub-tasks ‘query_attribute’ and ‘equal_attribute’ with attributes *color*, *shape*, *material*, and *size*. The two types of questions differ by answer type $y \in \mathcal{Y}$:

- **Wh-questions** (Wh-q): Questions about the *attribute* of an object, e.g., “What is the material of the large object...?”, where $y \in \{blue, cube, small, \dots, metal\}$ spans over $|color| = 8$, $|shape| = 3$, $|size| = 2$ and $|material| = 2$ (in total $|\mathcal{Y}| = 15$).

- **Yes/No questions (Y/N-q):** Questions that *compare* objects with respect to an attribute, e.g., “Does the cyan ball have the same material as ...?”, with $y \in \{yes, no\}$ (in total $|\mathcal{Y}| = 2$).

Task Order We learn Task A followed by Task B (TASKA \rightarrow TASKB), but experiment with *both* directions, i.e., by first assigning Wh-q to Task A and Y/N-q to Task B, and vice versa. We expect that the inherent difficulty of a task and the order in which tasks are learned have an impact on CL.

Single-head Evaluation CL methods can be tested in two ways. We opt for a *single-head* evaluation setup (see Fig. 7.1, lower) with an output space over labels for all tasks (here: all CLEVR labels). In contrast, in a *multi-head* setup predictions are restricted to task labels, as the task identifier is provided. Single-head is more difficult yet more realistic (Chaudhry et al., 2018).

7.3 Models and Experiments

VQA Model We take the model proposed by Yang et al. (2016) as a starting point, using the code released by Johnson et al. (2017d) (LSTM+CNN+SA). Questions are encoded with an RNN with LSTM units. Images are encoded with a ResNet-101 CNN pre-trained on ImageNet (He et al., 2016). The two representations are combined using Spatial Attention (SA) (Yang et al., 2016) to focus on the most salient objects and properties in the image and text. The final answer distribution is predicted with a MLP.

Baselines In order to measure catastrophic forgetting, we first consider per-task baselines: a random baseline (i.e., random stratified sample of the label distribution per task) and the results of a model trained independently on each task (i.e., over task-specific \mathcal{Y}). For CL, we report again a random baseline (this time a random stratified sample drawing predictions according to the answer distribution of both tasks), and we consider the *Naive* and *Cumulative* baselines proposed by Maltoni and Lomonaco (2018). The *Naive*

model is fine-tuned across tasks: It is first trained on Task A and then on Task B starting from the previously-learned parameters. The *Cumulative* model is trained from scratch on the training sets of both Task A and Task B. The performance of this model is a kind of upper bound, since it is the performance that a CL model should try to achieve.

Continual Learning Models In CL there are two broad families of methods: those that assume memory and access to explicit previous knowledge (instances), and those that have only access to compressed knowledge, such as previously-learned parameters. These two families correspond to rehearsal and regularization, respectively. A widely-used regularization-based approach is *Elastic Weight Consolidation (EWC)* (Kirkpatrick et al., 2017). A regularization term, parametrized by λ , is added to the loss function aiming the model to converge to parameters where it has a low error for both tasks. In the *Rehearsal* approach (Robins, 1995), the model is first trained on Task A, then the parameters are fine-tuned through batches taken from a dataset containing a small number of examples of Task A and the training set of Task B. The selection of training examples of Task A is done through uniform sampling.

Data and Training Details Since CLEVR has no published ground-truth answers for the test set, we split the original validation set into a validation and a test set. In order to avoid performance impact due to different training data sizes, we down-sample the training sets to the same size (Y/N-q data size), resulting in 125,654 training instances per task. The validation and test sets contain, respectively, 26,960 and 26,774 data points for Wh-q and 13,417 and 13,681 data points for Y/N-q.

For the baselines, we select the model which reaches maximum accuracy on the validation set of each task. For CL, we choose the model with the highest CL score, which is computed according to the validation set of each task pair. Details on the chosen hyper-parameters and evaluation metrics are provided in the appendix.

7.4 Results and Analysis

The main results are provided in Table 7.1. There are several take-aways.

Task Difficulty The results of the per-task models (cf. first two rows in Table 7.1) show that there is a large performance gap between the two tasks. Wh-q is easier (.81) than Y/N-q (.52), regardless of the fact that a priori the latter should be easier (as shown by the respective task-specific random baselines). The Y/N-q task-specific model performs only slightly above chance (.52, in line with what Johnson et al. (2017a) report for ‘equal attribute’ questions). This shows that despite the limited output space of the Y/N-q task, such type of questions in CLEVR are complex and require reasoning skills (Johnson et al., 2017a).

Catastrophic Forgetting We observe that extreme forgetting is at play. *Naive* forgets the previously learned skill completely: When tested on Task A after having been fine-tuned on Task B, it achieves 0.0 accuracy on the first task *for both directions* (I and II, cf. Table 7.1 lower). The *Cumulative* model by nature cannot forget, since it is trained on both tasks simultaneously, achieving .81 and .74 on Wh-q and Y/N-q, respectively. Interestingly, we observe an *asymmetric synergetic effect*. Being exposed to the Wh-q task helps the *Cumulative* model improve on Y/N-q, reaching results beyond the task-specific model (from .52 to .74). The effect is not symmetric as the accuracy on Wh-q does not further increase.

Does CL Help? Current CL methods show only limiting (or no) effect. *EWC* performs bad overall: In the II) setup (Y/N→WH, harder task first), *EWC* does not yield any improvement over the *Naive* model; in the WH→Y/N setup, the model’s result on Task A is above chance level (.25 vs. .04) but far off per-task performance (.81). The *Rehearsal* model forgets less than *Naive* and *EWC* in both setups: In the Y/N→WH setup, it is above chance level (.51 vs. .25) reaching per-task random baseline results on Y/N questions (i.e., the model is able to identify Task A, despite the harder single-head

Random (per-task)	WH: 0.09		Y/N: 0.50	
LSTM+CNN+SA	WH: 0.81		Y/N 0.52	
<hr/>				
CL SETUPS:	I) WH \rightarrow Y/N		II) Y/N \rightarrow WH	
	Wh	Y/N	Y/N	Wh
Random (both tasks)	0.04	0.25	0.25	0.04
Naive	0.00	0.61	0.00	0.81
EWC	0.25	0.51	0.00	0.83
Rehearsal	0.75	0.51	0.51	0.80
Cumulative	0.81	0.74	0.74	0.81

Table 7.1: Mean accuracy over 3 runs: Trained on each task independently (first two rows; per-task label space \mathcal{Y}) vs. CL setups (single-head label space over all \mathcal{Y}).

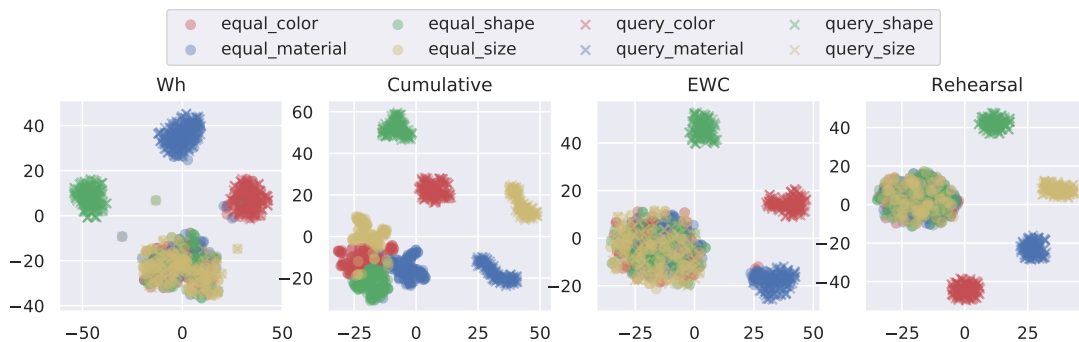


Figure 7.2: Analysis of the neuron activations on the penultimate hidden layer for the I) WH \rightarrow Y/N setup. “equal_{shape,color,material,size}” refers to Y/N-q, “query_{..}” refers to WH-questions.

setting, in contrast to the *Naive* and *EWC* models). There is no boost derived from being exposed to the Wh-q task in any of the two setups.

Task Order The results in Table 7.1 show that *the order of tasks plays an important role*: WH \rightarrow Y/N facilitates CL more than the opposite order: we notice a higher performance on the second task and less forgetting on the first task when WH is learned first. This confirms psycholinguistic evidence. Overall, *Rehearsal* works better than *EWC*, but mitigates forgetting only to a limiting degree.

Analysis In order to get a deeper understanding of the models, we analyze the penultimate hidden layer on a sample of 512 questions from the test sets of both tasks (cf. Fig. 7.2) and relate the representations to confusion matrices of the whole test sets (pro-

vided in the appendix) and test results (Table 7.1).

First of all, the model which has been trained on Wh-q is able to discriminate Wh-questions about different attributes very well. That is reflected in an overall high accuracy (.81). It otherwise clusters all instances from the other task (Y/N-q, which it has not been trained on) around Wh-questions related to size.

The *Cumulative* model, in contrast, is able to further tease the different kinds of Y/N questions apart. Questions about different attributes become distinguishable in the plot, although overall Y/N questions remain closer together than the clusters for Wh-q. This is in line with the lower performance of *Cumulative* on Y/N-q. Our examination of the confusion matrices confirms that the two question types are never confused by the *Cumulative* model. In contrast, the *Naive* model is very prone to this type of mistake.

As for the CL models, Fig. 7.2 (two rightmost plots) shows that *EWC* learns representations which are rather similar to those learned by the model trained on Wh-q independently: Y/N questions result in a big hard-to-distinguish “blob”, and are confused with Wh-q about size, as visible in Fig. 7.2 and the confusion matrix analysis (in the appendix). In contrast, *Rehearsal* remembers how to distinguish among all kinds of Wh-q *and* between Wh-q and Y/N-q. The error analysis confirms that the model hardly makes any mistakes related to task confusion. However, despite the higher performance than *EWC*, *Rehearsal* is still not able to discern well between different kinds of Y/N-q.

7.5 Related Work

Early work on life-long learning [Chen et al. \(2015\)](#); [Mitchell et al. \(2015\)](#) is related to ours, but typically concerns a single task (e.g., relation extraction). [Lee \(2017\)](#) aims to transfer conversational skills from a synthetic domain to a customer-specific application in dialogue agents, while [Yogatama et al. \(2019\)](#) show that current models for different Natural Language Processing tasks do not properly reuse previously-learned knowledge.

CL has been mostly studied in Computer Vision. To the best of our knowledge, little has been done on forgetting in the context of VQA. A study on forgetting in the VQA

which is the closest to ours is [Perez et al. \(2018\)](#). They show that their model forgets after being fine-tuned on data including images with objects of colors other than those previously seen. We took this work as starting point and extended it in order to consider different types of questions and to test different CL methods beyond fine-tuning.

7.6 Conclusion

We assessed to what extent training models on a curriculum of increasingly complex tasks improves transfer learning and mitigates catastrophic forgetting in the context of VQA. In order to train the model on tasks having different difficulty, we built two tasks involving different linguistic characteristics which are known to be learned sequentially by children and on which multimodal models reach different performance.

We show that the *order* in which models learn tasks is important, WH→Y/N facilitates continual learning more than the opposite order, thereby confirming psycholinguistic evidence. Moreover, our results show that dramatic forgetting is at play in VQA, and we empirically found *Rehearsal* to work better than a regularization-based method (*EWC*).

Moreover, our error analysis highlights the importance of taking the kind of mistakes made by the models into account: ideally, a model which does not detect Task A after having been exposed to Task B should be penalized more than a model that answers Task A with wrong task-related labels, but is still capable of identifying the task. Most importantly, our study revealed that differences in the inherent difficulty of the tasks at hand can have a strong impact on CL. Regularization-based methods like *EWC* seem to work less well when they are applied to tasks with different levels of difficulty, as in our experiments. We reserve a deeper investigation of this aspect to future research.

7.7 Summary

So far, we discovered that:

-
- + Training a neural multimodal model on a curriculum of increasingly complex tasks improves transfer learning and mitigates catastrophic forgetting;
 - However, dramatic forgetting is still at play in neural multimodal models.

Chapter 8

Conclusion

In this thesis, we investigated to what extent pre-trained Transformers effectively leverage pre-training and attention mechanisms, two of their major strengths, while dealing with multimodal tasks requiring to merge different pieces of information.

In the first place, we focused on the ability of pre-trained Transformers to detect salient information in either language modality or vision modality while grounding one into the other. In order to investigate that, we compared the performance of pre-trained Transformers, in a GuessWhat?! visual dialogue task, with the performance of Transformers trained from scratch and other not-attention-based models. Models receiving both the linguistic and visual modalities or the linguistic modality only (textual input). In this regard, the main outcomes of our research were that pre-trained Transformers greatly outperformed other models mostly due to their pre-training when they had to deal with referential guessing games. Indeed, they detect the most salient information and understand the structure of questions/answers in the dialogue history and identify regions in questions about an object in the image. Moreover, pre-trained Transformers were able to identify the most salient information in the dialogue history through their attention mechanism regardless of the order of the dialogue turns. However, the attention mechanism, at least if not associated with pre-training, did not bring a huge gain in accuracy, since non pre-trained Transformers did not outperform not-attention-based models trained from scratch. Although pre-trained Transformers reached a high perfor-

mance with respect to the other models, they struggled to obtain the same performance when compared to humans. We noticed that they were not able to effectively understand negative answers in a dialogue history in order to select the object the dialogue referred to. Moreover, they were not able to leverage their attention mechanism when answering spatial questions involving groups of objects in the image.

In the second place, we focused on the ability of pre-trained Transformers to combine complementary information coming from language and vision. In order to do that, first, we built an ad hoc task which requires to guess the most plausible action based on complementary linguistic and visual cues. Then, we compared the performance of pre-trained Transformers with the performance of Transformers trained from scratch and other not-attention-based models on the ad hoc built task. The previously-noticed pattern emerged in this study too: pre-trained Transformers outperformed other models mostly due to their pre-training. However, their performance is still weak compared to humans. This shows that state-of-the-art multimodal models still need to be improved in order to effectively integrate language and vision.

In our research, we were also interested in mitigating the issue of catastrophic forgetting, which affects both classical computational models and pre-trained Transformers. Indeed, ideally, models should be not only able to transfer their knowledge to new tasks, but also to avoid forgetting how to solve previously-solved tasks after having learned a new task. We investigated this parallel issue by assessing whether a better control of tasks' order helps to mitigate catastrophic forgetting in visual question answering tasks involving polar and Wh questions about images. We proved that training models on a curriculum of increasingly complex tasks mitigate forgetting and improves transfer. However, dramatic forgetting is still at place in neural multimodal models.

8.1 Contribution and Perspectives

Generally, our work aimed to assess whether pre-trained multimodal models are as “good” as they seem for merging language and vision. We believe that a better understanding of the way models integrate the two modalities is essential for a future

improvement of the performance of pre-trained models. In order to study their strengths and weaknesses, we proposed:

- A new setting to evaluate the ability of computational models to deal with negation in visual dialogue. Since models seem to struggle with negation, the designed setting may be a useful tool to going deeper on this crucial issue;
- A classification of spatial questions about objects in images which enables to identify strengths and weaknesses of computational models (e.g., identifying spatially single entities in groups in our experiment). This classification can be further used to investigate how different models or the same models in different conditions deal with spatial information;
- A task and a dataset to evaluate the ability of these models to integrate different modalities (language and vision in our task) that convey different but complementary information;
- An adaptation of pre-trained Transformers to solve referential visual tasks (identifying or answering questions about a target object in an image).

This research has many future directions:

- More research should be done on the issue to better understand whether computational models can learn to ground negation. Consequently, working on solving this crucial problem is needed. Moreover, given that models learn to encode negation in grounded contexts, the next step will be to transfer such skills to language-only settings by exploiting transfer learning methods;
- Since pre-trained models struggle to perform comparably to humans, an evaluation of other multimodal encoders may contribute to the development of better multimodal systems.

Finally, a less direct but crucial future direction of this study is to investigate how agents having different background knowledge adapt their language in order to improve their

knowledge by learning from each other. We are currently working on developing these abilities.

Chapter 9

Appendix

9.1 Chapter 3

In our experiments, we used the GuessWhat?! dataset (<http://guesswhat.ai/download>). The dataset contains 155000 English dialogues about approximately 66000 different images. The training split contains 108000 datapoints, the validation split 23000 datapoints, and the test split 23000 datapoints. We considered only the dialogues corresponding to the games succeeded by humans and having less or equal than 10 turns.

For training LSTM based models we adapted the source codes available at <https://github.com/shekharRavi/Beyond-Task-Success-NAACL2019> and at <https://github.com/GuessWhatGame/guesswhat/>. For training transformer based models we adapted the source code available at <https://github.com/huggingface/transformers>. The scripts for all the experiments and the modified models will be made available upon acceptance. For all models, we used the same hyperparameters of the original works. When adapting Transformers to the GuessWhat?! task, we scaled the representation of the CLS token from 768 to 512. We used PyTorch 1.0.1 for all models except for LSTM, for which we have used Tensorflow 1.3. All models are trained with Adam optimizer. For transformer based models we used a batch size equal to 16, a weight decay equal to 0.01, gradient clipping equal to 5, and a learning rate

which is warmed up over the first 10% iterations to a peak value of 0.00001 and then linearly decayed.

Regarding the infrastructure, we used 1 Titan V GPU. LSTM based models took about 15 hours for completing 100 training epochs. Transformer based models took about 4 days for completing 25 training epochs. Each experiment took about 10 minutes to evaluate the best trained models.

Details on the best epoch, the validation accuracy, and the number of parameters of each model are reported in Table 9.1.

Model	Best epoch	Validation accuracy	Parameters
LSTM	19	65.6	5,030,144
RoBERTa	7	68.7	125,460,992
RoBERTa-S	14	64.7	125,460,992
V-LSTM	9	65.2	10,952,818
LXMERT-S	16	65.2	208,900,978
LXMERT	12	70.0	208,900,978

Table 9.1: Epoch, validation accuracy, and number of parameters for best models.

9.2 Chapter 5

9.2.1 Game examples

It seems that RoBERTa takes spatial questions into account more than LXMERT, maybe because it exploits the spatial coordinates of the candidate objects whereas LXMERT overrides that information with the one it receives from the visual features. An example of dialogue where RoBERTa shows its strength on spatial questions is shown in figure 9.1. In this dialogue, models receive only the last turn and RoBERTa successfully exploits the spatial information of the last turn to correctly guess the fork which is the closest candidate to the camera, whereas LXMERT selects the wrong target.

LXMERT seems, instead, to shine when grounding questions which involve to recognize objects in the image. For instance, figure 9.2 shows an example of game where LXMERT correctly guesses the baby on the right, but RoBERTa does not.



Figure 9.1: Example of game where RoBERTa correctly guesses the object, but LXMERT does not.

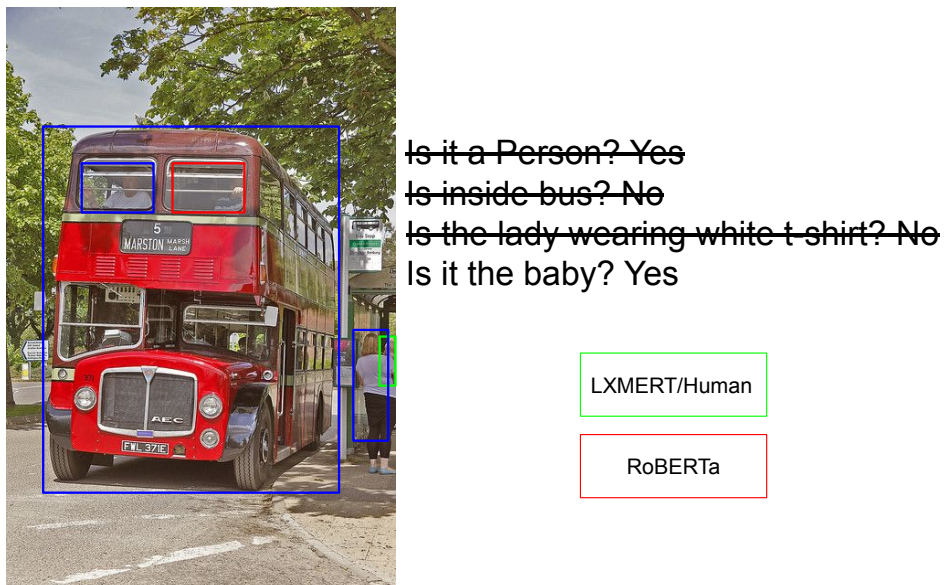


Figure 9.2: Example of game where LXMERT correctly guesses the object, but RoBERTa does not.

As highlighted in the experiments, it seems that the mistakes done by RoBERTa are more human-like. An example pointing out this phenomenon is shown in figure 9.3, where models and one human receiving the whole dialogue except for the last turn selects the wrong target in the image. Reading the dialogue, it is clear that the target is not on the bike on the right side of the image. Nevertheless, LXMERT wrongly guesses the person on that bike. Roberta and one human, instead, guess the person standing on the left part of the image as the target, but they are wrong since the dialogue specifies that the target is on the bike.



Figure 9.3: Example of game where models and humans are all wrong, but RoBERTa does the same mistake performed by one human.

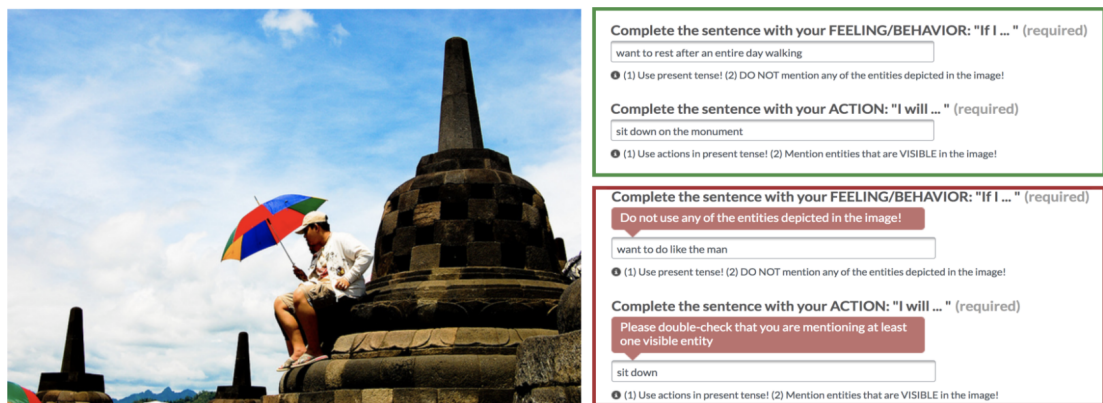


Figure 9.4: Data collection. Examples of good/bad annotations provided to participants at the beginning of the task. Note that the errors and corresponding warnings are shown to make them familiarize with the tool.

9.3 Chapter 6

9.3.1 Further Details on Data (Sec. 3)

Data Collection

Crowdsourcers are presented with detailed instructions and examples before starting with the annotation task. First, we introduce the task and provide them with some details to familiarize with the annotation tool. Then, we give them instructions regarding the constraints to be observed, i.e., for intentions: (1) to use present tense and (2) do


not mention any of the entities depicted in the image; for actions: (1) to use present tense and (2) do mention entities that are visible in the image. To make instructions and constraints clearer, we show them several examples of good/bad annotations (see Figure 9.4). Moreover, to make sure participants are performing the task properly (and, crucially, to avoid collecting fake data from automatic bots), a verification question is asked at the beginning of each image’s annotation phase. The verification question has multiple correct answers, and only by picking one of these answers participants can proceed with the annotation phase.

In addition, we add two sanity checks to the collected intentions. We check that (1) they have a length of at least 5 tokens; if this is not the case, participants are shown a warning and asked to fix their sentence; (2) they do not contain any noun referring to an entity that is grounded in the image; this is checked by means of a simple heuristic which extracts all the nouns from a given image’s MS-COCO captions. Nouns with a frequency of 2 occurrences or more are not allowed, and when typing them turkers are warned to modify their sentence.

BD2BB Dataset Statistics

As described in the chapter, the final BD2BB dataset includes 10,265 samples, where each sample includes a $\langle image, intention, target_action \rangle$ triple associated with 4 selected decoy actions. These triples were provided by 430 unique annotators. In particular, 253 were from the USA, 111 from the United Kingdom, 53 from Canada, 6 from Ireland, 5 from New Zealand, 2 from Australia. Each of them provided, on average, 23.87 $\langle image, intention, target_action \rangle$ tuples contained in the dataset (min 1, max 192).

Each sample contains 5 actions. On average, these actions were provided by 4.90 unique annotators (min 3, max 5); moreover, they were collected for 4.96 (min 3, max 5) unique images, i.e., decoy actions in each sample refer to different images than the target one in most of the cases.



Select one HUMAN entity from the list: (required)

- woman
- tennis
- player
- ball
- racket

Complete the sentence with your FEELING/BEHAVIOR: "If I ... " (required)

ⓘ (1) Use present tense! (2) DO NOT mention any of the entities depicted in the image!

Complete the sentence with your ACTION: "I will ... " (required)

ⓘ (1) Use actions in present tense! (2) Mention entities that are VISIBLE in the image!

Figure 9.5: Data collection. One annotation sample presented to participants. Given an image, participants are asked to provide an intention and an action. To ensure they are doing the task properly, a verification question is asked preliminarily. Answering the question correctly (multiple correct answers) leads to the annotation phase.

Meta-Annotation

Topics We manually inspected the 60 clusters obtained through k-means clustering and removed 6 clusters for which we could not identify a coherent topic. Examples of the action for each of the remaining 54 clusters with their corresponding label we have assigned are provided in Table 9.2.

Numeric 4-Code Annotation We organize our data through a two-step system of *wordcodes* using codes to mark the syntactic class and the word-type.

With the Stanford NLP parser [Chen and Manning \(2014\)](#) we extract from each action syntactic information and mark: 1) the main verb: “code1”; 2) the direct or indirect object of the main verb, as well as other complements related to the main verb: “code2”; 3) the second verb - if present (i.e. the verb of the coordinated or subordinated sentence): “code3”; 4) the object of the second verb - if present: “code4”. In this case, we considered not only the direct object of the second verb but also all the words referring to an object grounded in the corresponding image, that specifies the action expressed by the sentence. This way, for each action in which it was possible, we have a word

labels	action example	code1	code2	code3	code4
tennis	grab my tennis racket firmly and hit the ball	grab	racket	hit	ball
food	grab some delicious food	grab	food		
cake	cut the cake	cut	cake		
snacks	purchase a hot dog	purchase	hotdog		
actions with ball	hit the ball as hard as i can	hit	ball		
skateboard 1	go skateboarding	go	skateboard		
bikes and motos	take a ride on the motorbike	ride	motorbike		
skateboard 4	pull off this skateboard trick	pull off	trick		
surf	grab my surfboard and join the woman	grab	surfboard	join	woman
phone	call someone for a chat	call	someone		
interact with people	join these people and talk	join	people	talk	
baseball 2	yell at the batter to distract him	yell	batter	distract	batter
sport audience	watch this game	watch	game		
approaching women	try to get the woman's attention	get	attention		
pizza	order a slice of pizza	order	pizza		
ski	use my ski poles judiciously	use	ski poles		
drink	i will drink my drink and watch people walk by	drink	drink	watch	people
kids	move the baby so i can use the computer	move	baby	use	computer
cooking	help those women to cook	help	women	cook	
videogames	grab an extra remote and join the game	grab	remote	join	game
pets	take a piece of cake and give it to the dog	take	cake	give	dog
clothing	wear my sun glasses	wear	glasses		
relax	i would look for a seat to rest	look for	seat		
umbrella	use the pink umbrella	use	umbrella		
urban activities	try to cross the street to investigate the trams	cross	street	investigate	trams
laptop	i will use that laptop the best way	use	laptop		
baseball 3	i will play as batter in a game of baseball	play	game		
baseball 1	watch a baseball game	watch	baseball game		
team sports	i play a soccer game	play	soccer		
frisbee 2	join a frisbee team	join	team		
birthday	i will sing happy birthday to the girl	sing	happy birthday		girl
water sports	grab my board and ride the waves	grab	board	ride	wave
photo	to go to the bathroom to get a selfie	go to	bathroom	get	selfie
zoo animals	ride an elephant	ride	elephant		
public transports	i will get on the bus and take a trip	get on	bus	take	trip
skateboard 2	will sit on the wall and watch the skateboarder	sit	wall	watch	skateboarder
frisbee 1	i will leave these men to play their little frisbee game	leave	men	play	frisbee
wii	play a wii game	play	wii		
bedtime	instead go into my room and lay down	go	room	lay	
manual work / hobbies	use the scissors to make oragmi	use	scissors	make	origami
animals farm	watch the man shear the sheep	watch	man	shear	sheep
good intentions	get the right job	get	job		
kite	enjoy watching the people fly their kites	enjoy		watch	people
horse riding	ride a horse	ride	horse		
toilet things	brush my teeth	brush	teeth		
skateboard 3	i will go to skate park	go	skatepark		
street scenes	stealthily unzip his backpack and take his possessions	unzip	backpack	take	possession
ski and snow	take off my shirt and do a big ski jump in front of her	take off	shirt	do jump	woman
snowboard	go snowboarding	go	snowboard		
airport	board that ancient plane	board	plane		
fruit	buy and eat a banana	buy	banana	eat	banana
haircut	use the hairdryer	use	hairdryer		
women and food	tell the girl i hope she enjoys her pizza	tell	girl	enjoy	pizza
reading	read the newspaper	read	newspaper		

Table 9.2: To each of the 54 clusters we have assigned a label that summarize its main topic as illustrated by the examples of the actions we report for each cluster. Each action has been annotated with codes to mark the verb (code1) or the complement object (code2) of the main sentence and the verb (code3) and the complements (code4) of the secondary sentence. The clusters are listed by their size (from the biggest to the smallest).

that underlines the link between the linguistic and the visual aspect of the annotation. In Table 9.2, for each action given as an example of the cluster we highlight the words corresponding to each of the four codes. Statistics about this meta-annotation are reported in Table 9.3.

Furthermore, for each topic cluster, we assign a numeric *wordcode* to each unique word-type in the 4 *syntactic classes* described above. In other words, each sentence is translated into a code composed of 4 numbers, each one representing a unique word in the corresponding *syntactic class*.¹ Illustrative examples are given in Table 9.4.²

9.3.2 Further Details on Experiment (Sec. 5)

Models

The number of parameters of each model is reported in Table 9.6. The number of parameters is the same both in models trained from scratch and in pre-trained ones. The validation accuracy and epoch of the best models for each one of the three runs are reported in Table 9.5. For each of the three runs, we consider the model obtaining the best validation accuracy. For each model, we report mean and standard deviation of the test accuracies obtained across the three runs.

Baseline Our baseline is inspired by Jabri et al. (2016), but we use Softmax instead of Sigmoid as the final activation function to compute a probability distribution over all the candidates and choose the best one. We consider a version receiving image, intention and actions (**baseline_{LV}**), a version receiving image and actions (**baseline_V**), and a version receiving intention and actions (**baseline_L**). We used PyTorch 1.4.0. Baseline models were run on a CPU and their training took 33 seconds per epoch on average. We used a batch size equal to 32. We performed a grid search over two hyperparameters: the size of the hidden layer receiving concatenated figures (we tried values 8192 and

¹In the case in which we choose to consider more than one object, we create a compositional code, using the '+' mark

²Here numbers are assigned randomly, just to provide a concrete example of our meta-annotation.

2048) and the dropout probability of zeroing elements of the input tensor right after the ReLU activation function (we tried values 0.0 and 0.5). The combination of parameters which led to the best validation accuracy was a hidden layer having size 8192 and a dropout probability of 0.0 corresponding to not having any dropout.

RoBERTa The RoBERTa_{BASE} model we used has 12 self-attention layers with 12 heads each. It uses three special tokens, namely CLS, which is taken to be the representation of the given sequence, SEP, which separates sequences, and EOS, which denotes the end of the input. For each of the 5 $\langle image, intention, action \rangle$ datapoints in the sample, RoBERTa encodes the input as a sequence composed by CLS, the intention, SEP, the action, and EOS. As in the original work, we use the representation corresponding to the CLS token to use the encoder in the downstream task. For RoBERTa we used PyTorch 1.0.1 and we started from the source code available at <https://github.com/huggingface/transformers>. Both when fine-tuning the pre-trained model and when training the model from scratch, we used a batch size equal to 32 with 8 gradient accumulation steps, thereby having a batch size equal to 256, a weight decay equal to 0.01, gradient clipping equal to 5, and a learning rate which is warmed up over the first 10% steps to a peak value of 0.00005 and then linearly decayed.

LXMERT The LXMERT model we used has a Object-Relationship Encoder and a Language Encoder which encode relationships between regions and relationships words, respectively, through a self-attention mechanism, and a Cross-Modality Encoder which encode relationships between regions and words and vice-versa through a cross-modal attention mechanism followed by a self-attention mechanism. The number of layers in the Language Encoder, Object-Relationship Encoder, and Cross-Modality Encoder are 9, 5, and 5, respectively. As in RoBERTa, LXMERT uses the special tokens CLS and SEP. Differently from RoBERTa, LXMERT uses the special token SEP both to separate sequences and to denote the end of the textual input. As in the original work, we use the representation corresponding to the CLS token to use the encoder in the downstream task. For RoBERTa we used PyTorch 1.0.1 and we started from the source code available at <https://github.com/airsplay/lxmert>. As with

RoBERTa, both when fine-tuning the pre-trained model and when training the model from scratch, we used a batch size equal to 32 with 8 gradient accumulation steps, thereby having a batch size equal to 256, a weight decay equal to 0.01, gradient clipping equal to 5, and a learning rate which is warmed up over the first 10% steps to a peak value of 0.00005 and then linearly decayed.

9.4 Chapter 7

9.4.1 Implementation details

All models were trained using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0005 and with a batch size of 64. We stopped the training of the models whenever their accuracy on the validation set did not increase for 3 times in a row. Word embeddings had a size of 300. RNNs had two hidden layers and LSTM cells had a size of 1024. MLPs had one hidden layer of size 1024. We used the implementation released by Johnson et al. (2017d) for the LSTM+CNN+SA architecture.

9.4.2 Hyperparameter search

For *EWC*, we searched for the best λ value among 100, 1000, 10000. For *Rehearsal*, we considered sampling size values of 100, 1000, 10000 training examples from Task A. We reported results for the models having the highest *CL score* computed according to the validation sets of both tasks. For *EWC*, the best model had $\lambda = 100$; for *Rehearsal*, the best model used 10000 training examples from Task A in both orders, $WH \rightarrow Y/N$ and $Y/N \rightarrow WH$.

9.4.3 Continual Learning Evaluation Measures

Besides standard *Accuracy* (*Acc*), we consider metrics that have been introduced specifically to evaluate continual learning. In general, there is not much agreement among

authors about the best metrics to evaluate continual learning models. Thus, [Díaz-Rodríguez et al. \(2018\)](#) propose a set of comprehensive metrics which allow to evaluate different factors of continual learning models, such as accuracy, forgetting, backward/forward knowledge transfer, memory overhead, and computational efficiency. In this chapter, we focus on evaluating accuracy and forgetting across tasks. First, the authors define a measure describing the overall behavior of continual learning models. In particular, for each measure i describing a particular aspect of a model, let c_i (where $c_i \in [0, 1]$) be its average value and s_i (where $s_i \in [0, 1]$) be its standard deviation across r runs. Let $w_i \in [0, 1]$ (where $\sum_i^C w_i = 1$) be the weight given to measure i . Then, the *CL score*, which measures the overall score of the model across tasks, is defined. Higher values are better and the measure lies in the range $[0, 1]$. Formally, it is computed as follows:

$$CL\ score = \sum_{i=1}^{|C|} w_i c_i$$

Let $R \in \mathbb{R}^{N \times N}$ be the train-test accuracy matrix, whose element $R_{i,j}$ is equal to the test accuracy on task j after having trained the model up to task i , where N is the number of tasks. In the evaluation of the *CL score*, we take the following measures into account:

- *Mean accuracy (Mean acc)* ([Díaz-Rodríguez et al., 2018](#)), which measures the overall accuracy of the model on the learned tasks. Higher values are better and the measure lies in the range $[0, 1]$. Formally, it is defined as:

$$Mean\ acc = \frac{\sum_{i \geq j}^N R_{i,j}}{\frac{N(N+1)}{2}}$$

- *Remembering (Rem)* ([Díaz-Rodríguez et al., 2018](#)), which measures how much the model remembers how to perform previously learned tasks. Higher values are better and the measure lies in the range $[0, 1]$. Formally, it is defined as:

$$Rem = 1 - |\min(BWT, 0)|,$$

where *Backward transfer (BWT)* allows to measure the influence that learning a task has on the performance of the previously learned tasks and it is formally

defined as:

$$BWT = \frac{N_{i=2}^N \sum_{j=1}^{i-1} (R_{i,j} - R_{j,j})}{\frac{N(N-1)}{2}}$$

- *Intransigence (Int)* (Chaudhry et al., 2018), which captures how much a model is regularized towards preserving past knowledge and as a consequence less capable of learning new tasks. Lower values are better and the measure lies in the range $[-1, 1]$. Formally, intransigence on the k -th task is defined as:

$$I_k = a_k^* - a_{k,k},$$

where $a_{k,k}$ denotes the accuracy on task k of the model trained sequentially up to task k and a_k^* denotes the accuracy on task k of the *Cumulative* model trained on tasks $1, \dots, k$. In the experiments, we only measure intransigence for the second task, because we take only two tasks into account and it does not make sense to compute intransigence for the first task. Hence, *Int* denotes I_2 .

CL score requires that each measure lies in the range $[0, 1]$ and that higher values are better. *Mean acc* and *Rem* already satisfy these constraints, whereas *Int* does not. Hence, when computing *CL score* in the case of *Int*, c_i is transformed to $c_i = 1 - (c_i + 1)/2$ to scale its range to $[0, 1]$ and to preserve the monotonicity of *CL score*.

9.4.4 Elastic Weight Consolidation

Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) is a regularization approach which introduces plasticity in artificial neural networks by slowing down learning in weights which are important to solve previously learned tasks. The method takes inspiration from the human brain, in which the plasticity of synapses which are important to solve previously learned tasks is reduced. *EWC* adds a regularization term to the loss function allowing the model to converge to parameters where it has a low error for both tasks. In particular, if Task A and Task B have to be learned sequentially *EWC*, after having learned Task A, computes the Fisher Information Matrix, whose i -th diagonal element assesses how important parameter i of the model is to solve Task A. Then,

the model is trained on Task B starting from the parameters previously learned to solve Task A by minimizing the following loss function:

$$L = L_B(\theta) + \frac{\lambda}{2} \sum_i F_{i,i}(\theta_i - \theta_i^A)^2,$$

where L_B is the loss function of Task B, $F_{i,i}$ is the i -th diagonal element of the Fisher Information Matrix, θ_i is the i -th parameter, θ_i^A is the optimal i -th parameter for Task A, and λ controls the regularization strength, i.e. the higher it is, the more it is important to remember Task A.

9.4.5 Confusion matrices

Tables 9.7, 9.8, 9.9, 9.10, and 9.11 show the confusion matrices of the Wh, *Naive*, *Cumulative*, *Rehearsal*, and *EWC* models, respectively, on the WH \rightarrow Y/N setup. Tables 9.7, 9.13, 9.14, 9.15, 9.16, instead, show the confusion matrices of the Y/N, *Naive*, *Cumulative*, *Rehearsal*, and *EWC* models, respectively, on the Y/N \rightarrow WH setup. In particular, predictions on these confusion matrices are grouped according to their category, so that rows represent the question type each question belongs to, columns represent the category each answer belongs to, and cells show the number of predictions the model obtains for a particular question type and answer category.

9.4.6 Neuron activations

Figures 9.6 and 9.7 show the neuron activations on the penultimate hidden layer of *Naive* model for the I) WH \rightarrow Y/N setup and the model trained independently on Y/N-q, respectively. All the visualizations of neuron activations reported in the experiments are obtained by computing the vectors containing the neuron activations of the penultimate hidden layer of the model during forward propagation and by plotting the resulting vectors transformed into two dimensions through *t-distributed Stochastic Neighbor Embedding (t-SNE)* Maaten and Hinton (2008).

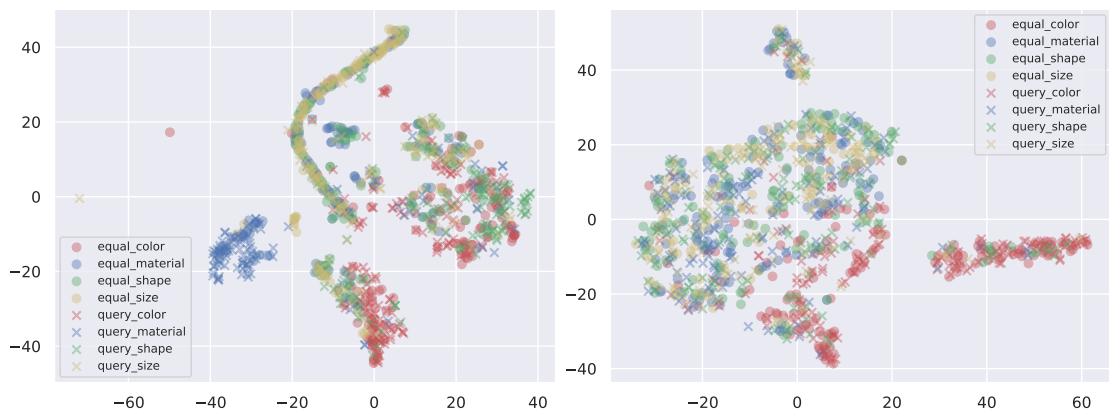


Figure 9.6: Analysis of the neuron activa- Figure 9.7: Analysis of the neuron activa-
 tions on the penultimate hidden layer of the tions on the penultimate hidden layer of the
Naive model for the I) $WH \rightarrow Y/N$ setup. model trained independently on $Y/N-q$.

labels	#actions	#code1	#code2	#code3	#code4
tennis	580	90	50	79	41
food	408	76	63	81	57
cake	334	60	37	65	74
snacks	316	68	82	26	50
actions with ball	298	71	27	54	34
skateboard 1	270	61	48	51	43
bikes and motos	269	86	55	59	51
skateboard 4	267	54	25	38	33
surf	262	66	50	52	22
phone	261	72	48	60	49
interact with people	261	66	58	62	22
baseball 2	259	82	42	69	30
sport audience	250	70	40	32	46
approaching women	227	84	54	49	70
pizza	226	43	23	37	42
ski	223	53	35	26	34
drink	222	53	46	50	39
kids	213	78	47	41	73
cooking	213	68	70	45	45
videogames	212	47	34	42	40
pets	202	80	47	44	32
clothing	202	54	61	48	47
relax	192	33	14	46	61
umbrella	186	56	24	32	26
urban activities	181	75	56	55	59
laptop	180	69	34	43	45
baseball 3	177	33	30	27	6
baseball 1	177	42	32	60	44
team sports	172	38	31	27	50
frisbee 2	172	25	25	29	22
birthday	170	62	71	46	59
water sports	165	87	60	38	41
photo	163	39	21	30	44
zoo animals	161	57	25	32	39
public transports	159	46	28	23	22
skateboard 2	158	45	36	35	25
frisbee 1	154	39	11	31	27
wii	149	36	22	35	22
bedtime	144	53	38	51	29
manual work / hobbies	139	69	75	44	60
animals farm	139	69	41	32	26
good intentions	132	66	64	44	32
kite	125	28	18	31	17
horse riding	118	49	22	22	29
toilet things	105	43	38	29	24
skateboard 3	98	22	16	18	14
street scenes	96	56	37	26	35
ski and snow	95	48	26	31	23
snowboard 1	94	27	26	21	17
airport	93	48	30	35	12
fruit	89	33	18	24	20
haircut	54	31	21	19	15
women and food	43	24	18	22	14
reading	32	11	11	11	7

Table 9.3: Statistics about the meta-annotation of the data. For each cluster, we report the number of actions, of verbs in the main sentence (code1) and in the secondary sentence (code3) and the number of nouns occurring as complements in the main sentence (code2) and in the secondary sentence (code4).

cluster	action	code1	code2	code3	code4
food	join the people in the restaurant to enjoy a meal	join 1	people 77	enjoy 15	meal 28
food	get some food with the people	get 107	food 6	0	people 666
frisbee	join this man playing frisbee	join 9	man 11	play 13	frisbee 14
frisbee	catch the frisbee and throw it again	catch 777	frisbee 777	throw 8	frisbee 14

Table 9.4: Examples of action with the word-type codes. Note that, (1) the same verb - e.g. *join*, line 1 and line 3 - in different clusters gets different codes; (2) the same object within the same cluster if in different syntactic positions (- e.g. *frisbee* in line 4), gets different codes but (3) the same object, in the same cluster, in the same syntactic position - e.g. *frisbee*, line 3 and line 4 - gets the same code.

Model	Run 1		Run 2		Run 3	
	Epoch	Valid. acc.	Epoch	Valid. acc.	Epoch	Valid. acc.
baseline _L	19	0.449	28	0.446	41	0.462
baseline _V	25	0.453	21	0.467	23	0.453
baseline _{LV}	22	0.481	34	0.496	36	0.480
RoBERTa _L ^s	3	47.1	2	46.8	2	47.1
LXMERT _V ^s	8	32.0	8	29.9	48	30.7
LXMERT _{LV} ^s	35	50.2	9	50.8	28	50.2
RoBERTa _L	12	0.571	36	0.557	38	0.550
LXMERT _V	38	0.593	49	0.588	31	0.592
LXMERT _{LV}	44	0.643	36	0.647	18	0.595

Table 9.5: Epoch and validation accuracy of the best models for each run.

Model	Number of parameters
baseline _L	4931585
baseline _V	19251201
baseline _{LV}	21708801
RoBERTa _L	124646401
LXMERT _V	194352385
LXMERT _{LV}	194352385

Table 9.6: Number of parameters of each model. The number of parameters is the same both in models trained from scratch and in pre-trained ones.

Wh	query_color	query_shape	query_size	query_material	Yes/No
query_color	6752	0	0	0	0
query_shape	0	6702	0	0	0
query_size	0	0	6666	0	0
query_material	0	0	0	6653	0
equal_color	1204	14	2088	3	0
equal_shape	26	1150	2232	2	0
equal_size	0	0	3430	0	0
equal_material	21	34	1440	2037	0

Table 9.7: Confusion matrix of the model trained independently on Wh-q.

Naive	query_color	query_shape	query_size	query_material	Yes/No
query_color	15	0	0	0	6738
query_shape	0	81	0	0	6621
query_size	0	0	0	0	6666
query_material	0	0	0	148	6505
equal_color	0	0	0	0	3309
equal_shape	0	0	0	0	3410
equal_size	0	0	0	0	3430
equal_material	0	0	0	0	3532

Table 9.8: Confusion matrix of the *Naive* model on the WH \rightarrow Y/N setup.

Cumulative	query_color	query_shape	query_size	query_material	Yes/No
query_color	6752	0	0	1	0
query_shape	0	6702	0	0	0
query_size	0	0	6665	1	0
query_material	0	0	0	6653	0
equal_color	0	0	0	0	3309
equal_shape	0	0	0	0	3410
equal_size	0	0	0	0	3430
equal_material	0	0	0	0	3532

Table 9.9: Confusion matrix of the *Cumulative* model on the WH \rightarrow Y/N setup.

Rehearsal	query_color	query_shape	query_size	query_material	Yes/No
query_color	6743	1	8	1	0
query_shape	0	6702	0	0	0
query_size	0	0	6664	0	2
query_material	1	0	1	6651	0
equal_color	0	0	0	0	3309
equal_shape	0	0	0	0	3410
equal_size	0	0	0	0	3430
equal_material	0	0	0	0	3532

Table 9.10: Confusion matrix of the best *Rehearsal* model on the WH \rightarrow Y/N setup.

EWC	query_color	query_shape	query_size	query_material	Yes/No
query_color	6715	0	0	1	37
query_shape	0	5479	0	0	1223
query_size	0	0	0	0	6657
query_material	0	0	0	1337	5316
equal_color	0	0	0	0	3309
equal_shape	0	2	0	0	3408
equal_size	0	0	1	0	3429
equal_material	0	0	0	0	3532

Table 9.11: Confusion matrix of the best *EWC* model on the $WH \rightarrow Y/N$ setup.

Y/N	query_color	query_shape	query_size	query_material	Yes/No
query_color	0	0	0	0	6753
query_shape	0	0	0	0	6753
query_size	0	0	0	0	6666
query_material	0	0	0	0	6653
equal_color	0	0	0	0	3309
equal_shape	0	0	0	0	3410
equal_size	0	0	0	0	3430
equal_material	0	0	0	0	3532

Table 9.12: Confusion matrix of the model trained independently on $Y/N-q$.

Naive	query_color	query_shape	query_size	query_material	Yes/No
query_color	6753	0	0	0	0
query_shape	0	6701	1	0	0
query_size	0	0	6666	0	0
query_material	1	0	1	6651	0
equal_color	2732	38	229	310	0
equal_shape	1317	1144	346	603	0
equal_size	1330	16	1559	525	0
equal_material	1297	0	30	2205	0

Table 9.13: Confusion matrix of the *Naive* model on the $Y/N \rightarrow WH$ setup.

Cumulative	query_color	query_shape	query_size	query_material	Yes/No
query_color	6753	0	0	0	0
query_shape	1	6701	0	0	0
query_size	0	0	6666	0	0
query_material	0	0	0	6653	0
equal_color	0	0	0	0	3309
equal_shape	0	0	0	0	3410
equal_size	0	0	0	0	3430
equal_material	0	0	0	0	3532

Table 9.14: Confusion matrix of the *Cumulative* model on the $Y/N \rightarrow WH$ setup.

Rehearsal	query_color	query_shape	query_size	query_material	Yes/No
query_color	6752	0	1	0	0
query_shape	0	6702	0	0	0
query_size	0	0	6666	0	0
query_material	1	0	1	6651	0
equal_color	0	0	0	0	3309
equal_shape	1	0	0	0	3409
equal_size	0	0	1	0	3429
equal_material	0	0	0	0	3532

Table 9.15: Confusion matrix of the best *Rehearsal* model on the Y/N \rightarrow WH setup.

EWC	query_color	query_shape	query_size	query_material	Yes/No
query_color	6748	4	0	1	0
query_shape	0	6701	1	0	0
query_size	0	0	6666	0	0
query_material	1	0	0	6652	0
equal_color	3110	9	17	173	0
equal_shape	801	1214	69	1326	0
equal_size	542	35	35	1674	2
equal_material	464	2	1	3065	0

Table 9.16: Confusion matrix of the best *EWC* model on the Y/N \rightarrow WH setup.

Bibliography

Shubham Agarwal, Trung Bui, Joon-Young Lee, Ioannis Konstas, and Verena Rieser. 2020. History for visual dialog: Do we really need it? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, pages 8182–8197.

Arjun R. Akula, Spandana Gella, Yaser Al-Onaizan, Song-Chun Zhu, and Siva Reddy. 2020. Words aren’t enough, their order matters: On the robustness of grounding visual referring expressions. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Gen Li and Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. 2020. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of AAAI*.

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert. 1991. The HCRC map task corpus. *Language and Speech* 34:351–366.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. 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.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018b. Bottom-up and top-down attention for image cap-

- tioning 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. 2015a. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2425–2433.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015b. VQA: Visual question answering. In *International Conference on Computer Vision (ICCV)*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .
- Jason Baldridge, Tania Bedrax-Weiss, Daphne Luong, Srini Narayanan, Bo Pang, Fernando Pereira, Radu Soricut, Michael Tseng, and Yuan Zhang. 2018. Points, paths, and playscapes: Large-scale spatial language understanding tasks set in the real world. In *Proceedings of the First International Workshop on Spatial Language Understanding*. Association for Computational Linguistics, New Orleans.
- Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* 41(2):423–443.
- Marco Baroni. 2016. Grounding distributional semantics in the visual world. *Language and Linguistics Compass* 10(1):3–13.
- Lisa Beinborn, Teresa Botschen, and Iryna Gurevych. 2018. Multimodal grounding for language processing. In *Proceedings of the 27th International Conference on Computational Linguistics*. pages 2325–2339.
- Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on* 5.

- Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2016a. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research* 55:409–442.
- Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2016b. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research* 55:409–442.
- Nilavra Bhattacharya, Qing Li, and Danna Gurari. 2019. Why does a visual question have different answers? In *Proceedings of the IEEE International Conference on Computer Vision*. pages 4271–4280.
- P. Carpenter and M. Just. 1975. Sentence comprehension: A psycholinguistic processing model of verification. *Psychological Review* 82:45–73.
- 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 for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pages 169–174.
- Wei-Lun Chao, Hexiang Hu, and Fei Sha. 2018. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. pages 431–441.
- 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 *Proceedings of the Fifth AAIL Conference on Human Computation and Crowdsourcing (HCOMP)*.
- Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*.

- Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 740–750.
- Xiaofan Chen, Songyang Lao, and Ting Duan. 2020. Multimodal fusion of visual dialog: A survey. In *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence*. pages 302–308.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019a. UNITER: Learning universal image-text representations. ArXiv:1909.11740.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019b. [UNITER: learning universal image-text representations](https://arxiv.org/abs/1909.11740). *CoRR* abs/1909.11740. <http://arxiv.org/abs/1909.11740>.
- Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015. Lifelong learning for sentiment classification. In *ACL*. Short paper.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- H. Clark and W. Chase. 1972. On the process of comparing sentences against pictures. *Cognitive Psychology* 3:472–517.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? An analysis of BERT’s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. pages 276–286.

- Guillem Collell and Marie-Francine Moens. 2018. [Learning representations specialized in spatial knowledge: Leveraging language and vision](#). *Transactions of the Association for Computational Linguistics* 6:133–144. <https://www.aclweb.org/anthology/Q18-1010>.
- R. Dale and N. Duran. 2011. The cognitive dynamics of negated sentence verification. *Cognitive Science* 35:983–996.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017a. Visual Dialog. In *CVPR*.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017b. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 326–335.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017c. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 326–335.
- Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. 2017d. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. In *2017 IEEE International Conference on Computer Vision*. pages 2951–2960.
- Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. GuessWhat?! Visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 5503–5512.
- Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. 2017a. Guesswhat?! visual object discovery through multi-modal dialogue. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pages 4466–4475.
- Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and

- Aaron C. Courville. 2017b. Guesswhat?! Visual object discovery through multi-modal dialogue. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pages 248–255.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4171–4186. <https://doi.org/10.18653/v1/N19-1423>.
- Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. 2018. Don't forget, there is more than forgetting: new metrics for continual learning. In *Workshop on Continual Learning, NeurIPS*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* .
- 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.
- Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847* .

- Zhe Gan, Yu Cheng, Ahmed El Kholy, Linjie Li, Jingjing Liu, and Jianfeng Gao. 2019. Multi-step reasoning via recurrent dual attention for visual dialog. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pages 6463–6474.
- Albert Gatt and Kees van Deemter. 2007. Incremental generation of plural descriptions: Similarity and partitioning. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*. pages 102–111.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*. PMLR, pages 1243–1252.
- Mehdi Ghanimifard and Simon Dobnik. 2017. Learning to compose spatial relations with grounded neural language models. In *IWCS 2017 - 12th International Conference on Computational Semantics - Long papers*.
- Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. In <http://web.archive.org/save/http://Skylion007.github.io/OpenWebTextCorpus..>
- Tejas Gokhale, Pratyay Banerjee, and Yezhou Yang. 2020. Vqa-lol: Visual question answering under the lens of logic. In *Proceedings of ECCV*.
- Diana Gonzalez-Rico and Gibran Fuentes Pineda. 2018. [Contextualize, show and tell: A neural visual storyteller](#). *CoRR* abs/1806.00738. <http://arxiv.org/abs/1806.00738>.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 6904–6913.

- Claudio Greco, Barbara Plank, Raquel Fernández, and Raffaella Bernardi. 2019. Psycholinguistics meets continual learning: Measuring catastrophic forgetting in visual question answering. *arXiv preprint arXiv:1906.04229*.
- Claudio Greco, Alberto Testoni, and Raffaella Bernardi. 2020. [Grounding dialogue history: Strengths and weaknesses of pre-trained transformers](#). In Matteo Baldoni and Stefania Bandini, editors, *AIXIA 2020 - Advances in Artificial Intelligence - XIXth International Conference of the Italian Association for Artificial Intelligence, Virtual Event, November 25-27, 2020, Revised Selected Papers*. Springer, volume 12414 of *Lecture Notes in Computer Science*, pages 263–279. https://doi.org/10.1007/978-3-030-77091-4_17.
- Dan Guo, Hui Wang, and Meng Wang. 2019. Dual visual attention network for visual dialog. In *IJCAI*. pages 4989–4995.
- Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018a. Vizwiz grand challenge: Answering visual questions from blind people. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018b. VizWiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3608–3617.
- Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. [The PhotoBook dataset: Building common ground through visually-grounded dialogue](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. <https://doi.org/10.18653/v1/P19-1184>.
- He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. pages 1766–1776.

- 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.
- Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. 2018. Generating counterfactual explanations with natural language. In *ICML Workshop on Human Interpretability in Machine Learning*. pages 95–98.
- Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- 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*. pages 19–28.
- Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47:853–899.
- Md Mosharaf Hossain, Antonios Anastasopoulos, Eduardo Blanco, and Alexis Palmer. 2020a. [It’s not a non-issue: Negation as a source of error in machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, pages 3869–3885. <https://doi.org/10.18653/v1/2020.findings-emnlp.345>.
- Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020b. [An analysis of natural language inference benchmarks through the lens of negation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, pages 9106–9118. <https://doi.org/10.18653/v1/2020.emnlp-main.732>.
- Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. [Understanding by understanding not: Modeling negation in language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

- Technologies*. Association for Computational Linguistics, Online, pages 1301–1312. <https://doi.org/10.18653/v1/2021.naacl-main.102>.
- Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1233–1239.
- Drew A Hudson and Christopher D Manning. 2019. Gqa: a new dataset for compositional question answering over real-world images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019a. Meet up! a corpus of joint activity dialogues in a visual environment. In *Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*. SEMDIAL, London, United Kingdom.
- Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019b. [Tell Me More: A Dataset of Visual Scene Description Sequences](#). In *Proceedings of the 12th International Conference on Natural Language Generation*. <https://www.aclweb.org/anthology/W19-8621>.
- Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan L Boyd-Graber, Hal Daumé III, and Larry S Davis. 2017. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 6478–6487.
- Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. 2016. Revisiting visual question answering baselines. In *European Conference on Computer Vision*. Springer, pages 727–739.
- Unnat Jain, Ziyu Zhang, and Alexander G Schwing. 2017. Creativity: Generating diverse questions using variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 6485–6494.

- Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. 2011. Learning cross-modality similarity for multinomial data. In *2011 International Conference on Computer Vision*. IEEE, pages 2407–2414.
- Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017a. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 2901–2910.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017b. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 2901–2910.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2017c. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*. volume abs/1612.06890.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017d. Inferring and executing programs for visual reasoning. In *ICCV*.
- Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 1965–1973.
- 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.
- Nora Kassner and Hinrich Schütze. 2020. [Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, pages 7811–7818. <https://doi.org/10.18653/v1/2020.acl-main.698>.

- E. L. Kaufman, M. Lord, T. Reese, and J. Volkmann. 1949. The discrimination of visual number. *The American journal of psychology* page 498–525.
- Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? A critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 5010–5015.
- John D. Kelleher and Simon Dobnik. 2017. What is not where: the challenge of integrating spatial representations into deep learning architectures. In *In CLASP Papers in Computational Linguistics. Proceedings of the Conference on Logic and Machine Learning in Natural Language*. pages 41–52.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](http://arxiv.org/abs/1412.6980). In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. <http://arxiv.org/abs/1412.6980>.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS* .
- Emiel Krahmer and Kees van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics* 38(1):173–218.
- Gunther R Kress. 2010. *Multimodality: A social semiotic approach to contemporary communication*. Taylor & Francis.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017a. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123(1):32–73.

- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017b. 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. *Advances in neural information processing systems* 25.
- Julia Kruk, Jonah Lubin, Karan Sikka, Xiao Lin, Dan Jurafsky, and Ajay Divakaran. 2019. Integrating text and image: Determining multimodal document intent in Instagram posts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. pages 4614–4624.
- 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, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361(10):1995.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.
- Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*, Springer, pages 9–48.
- Sungjin Lee. 2017. Toward continual learning for conversational agents. In *ACL*.
- Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. TVQA: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 1369–1379.

- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A simple and performant baseline for vision and language. ArXiv:1908.03557.
- Yining Li, Chen Huang, Xiaoou Tang, and Chen Change Loy. 2017. Learning to disambiguate by asking discriminative questions. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 3419–3428.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014a. Microsoft COCO: Common objects in context. In *Proceedings of ECCV (European Conference on Computer Vision)*. pages 740–755.
- 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.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. **RoBERTa: A Robustly optimized BERT pretraining approach**. *CoRR* abs/1907.11692. <http://arxiv.org/abs/1907.11692>.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoERTa: A robustly optimized bert pretraining approach. ArXiv:1907.11692.
- J. Lønning. 1997. Plurals and collectivity. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, Elsevier, pages 1009–1054.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019a. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*. pages 13–23.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019b. ViLBERT: Pretraining

- task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*. pages 13–23.
- Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, De Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of CVPR*.
- Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 375–383.
- Stephanie Lukin, Reginald Hobbs, and Clare Voss. 2018. A pipeline for creative visual storytelling. In *Proceedings of the First Workshop on Storytelling*. pages 20–32.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.
- Davide Maltoni and Vincenzo Lomonaco. 2018. Continuous learning in single-incremental-task scenarios. *arXiv preprint arXiv:1806.08568* .
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60.
- Mauricio Mazuecos, Alberto Testoni, Raffaella Bernardi, and Luciana Benotti. 2020. On the role of effective and referring questions in GuessWhat?! In *Proceedings of the First Workshop on Advances in Language and Vision Research*. Association for Computational Linguistics, Online, pages 19–25.
- James L McClelland, Bruce L McNaughton, and Randall C O’reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychol. Review* 102(3).
- Harry McGurk and John MacDonald. 1976. Hearing lips and seeing voices. *Nature* 264(5588):746–748.

- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.
- T. Mitchell, W. Cohen, E. Hruscha, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohammad, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *AAAI*.
- Sara Moradlou and Jonathan Ginzburg. 2016. Young children’s answers to questions. In *Workshop on the Role of Pragmatic Factors on Child Language Processing*.
- Sara Moradlou, Xiaobei Zheng, Ye Tian, and Jonathan Ginzburg. 2018. Wh-questions are understood before polars. In *Proceedings of Architectures and Mechanisms for Language Processing (AMLaP)*.
- Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. pages 462–472.
- 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)*. volume 1, pages 1802–1813.
- Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. 2019a. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. *arXiv preprint arXiv:1912.02379*.
- Vishvak Murahari, Prithvijit Chattopadhyay, Dhruv Batra, Devi Parikh, and Abhishek Das. 2019b. [Improving generative visual dialog by answering diverse questions](#). In

- Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. pages 1449–1454. <https://www.aclweb.org/anthology/D19-1152>.
- Sebastian Nagel. 2016. Cc-news. In . <http://web.archive.org/save/http://commoncrawl.org/2016/10/news-dataset-available/>.
- Ann Nordmeyer and Michael C. Frank. 2014. A pragmatic account of the processing of negative sentences. *Cognitive Science* 36.
- Mike Oaksford. 2002. Contrast classes and matching bias as explanations of the effects of negation on conditional reasoning. *Thinking & reasoning* 8(2):135–151.
- Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems* 24:1143–1151.
- Wei Pang and Xiaojie Wang. 2020. Visual dialogue state tracking for question generation. In *Proceedings of 34th AAAI Conference on Artificial Intelligence*.
- Ivandr  Paraboni, Alex Gwo Jen Lan, Matheus Mendes de Sant’Ana, and Fl vio Luiz Coutinho. 2017. Effects of cognitive effort on the resolution of overspecified descriptions. *Computational Linguistics* 43(2):451–459.
- Ivandr  Paraboni, Kees van Deemter, and Judith Masthoff. 2007. Generating referring expressions: Making referents easy to identify. *Computational Linguistics* 33(2):229–254.
- Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. In *31st IEEE Conference on Computer Vision and Pattern Recognition*.
- Sarah Partan and Peter Marler. 1999. Communication goes multimodal. *Science* 283(5406):1272–1273.

- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. <http://www.aclweb.org/anthology/D14-1162>.
- Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *AAAI*.
- Sandro Pezzelle, Claudio Greco, Greta Gandolfi, Eleonora Gualdoni, and Raffaella Bernardi. 2020. Be different to be better! a benchmark to leverage the complementarity of language and vision. In *Findings of the association for computational linguistics: EMNLP 2020*. pages 2751–2767.
- M. Piazza, A. Mechelli, B. Butterworth, and C.J. Price. 2002. Are subitizing and counting implemented as separate or functionally overlapping processes? *NeuroImage* pages 435–446.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Online, pages 101–108.
- Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. 2019. [Sunny and dark outside?! Improving answer consistency in VQA through entailed question generation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 5860–5865. <https://doi.org/10.18653/v1/D19-1596>.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. pages 91–99.
- Mark Ring. 1997. CHILD: A first step towards continual learning. *Machine Learning* 28(1).

- Anthony Robins. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science* 7(2):123–146.
- Sebastian Ruder. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. thesis, National University of Ireland.
- David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.
- David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5(3):1.
- Lee Sang-Woo, Gao Tong, Yang Sohee, Yao Jaejun, and Ha Jung-Woo. 2019. Large-scale answerer in questioner’s mind for visual dialog question generation. In *Proceedings of International Conference on Learning Representations, ICLR*.
- Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. [Do neural dialog systems use the conversation history effectively? An empirical study.](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pages 32–37. <https://www.aclweb.org/anthology/P19-1004>.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45(11):2673–2681.
- Himanshu Sharma and Anand Singh Jalal. 2021. A survey of methods, datasets and evaluation metrics for visual question answering. *Image and Vision Computing* page 104327.
- 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*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pages 1218–1233. <https://www.aclweb.org/anthology/C18-1104>.

- Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017a. FOIL it! Find one mismatch between image and language caption. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 255–265.
- Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017b. FOIL it! Find One mismatch between Image and Language caption. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 255–265.
- Ravi Shekhar, Aashish Venkatesh, Tim Baumgärtner, Elia Bruni, Barbara Plank, Raffaella Bernardi, and Raquel Fernández. 2019. [Beyond task success: A closer look at jointly learning to see, ask, and GuessWhat](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pages 2578–2587. <https://doi.org/10.18653/v1/N19-1265>.
- Kevin J Shih, Saurabh Singh, and Derek Hoiem. 2016. Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 4613–4621.
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Amanpreet Singh, Vedanuj Goswami, and Devi Parikh. 2020. [Are we pretraining it right? Digging deeper into visio-linguistic pretraining](#). *CoRR* abs/2004.08744. <https://arxiv.org/abs/2004.08744>.
- Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics* 2:207–218.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*.
- 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>.
- Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pages 6418–6428.
- Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. pages 2440–2448.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014a. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014b. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.
- Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. 2020. [Investigating transferability in pretrained language models](#). *CoRR* abs/2004.14975. <https://arxiv.org/abs/2004.14975>.
- Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. pages 5103–5114.
- Kento Terao, Toru Tamaki, Bisser Raytchev, Kazufumi Kaneda, and Shun’ichi Satoh. 2020. [Which visual questions are difficult to answer? Analysis with entropy of answer distributions](#). *CoRR* abs/2004.05595. <https://arxiv.org/abs/2004.05595>.

- Alberto Testoni, Claudio Greco, and Raffaella Bernardi. 2021. Artificial intelligence models do not ground negation, humans do. guesswhat?! dialogues as a case study. *Frontiers in big Data* 4.
- Alberto Testoni, Claudio Greco, Tobias Bianchi, Mauricio Mazuecos, Agata Marcante, Luciana Benotti, and Raffaella Bernardi. 2020. They are not all alike: Answering different spatial questions requires different grounding strategies. In *Proceedings of the Third International Workshop on Spatial Language Understanding*. pages 29–38.
- Alberto Testoni, Ravi Shekhar, Raquel Fernández, and Raffaella Bernardi. 2019. The devil is in the detail: A magnifying glass for the GuessWhich visual dialogue game. In *Proceedings of the 23rd SemDial Workshop on the Semantics and Pragmatics of Dialogue (LondonLogue)*. pages 15–24.
- Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847* .
- Tao Tu, Qing Ping, Govindarajan Thattai, Gokhan Tur, and Prem Natarajan. 2021. Learning better visual dialog agents with pretrained visual-linguistic representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 5622–5631.
- Takuma Udagawa and Akiko Aizawa. 2019. A natural language corpus of common grounding under continuous and partially-observable context. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 33, pages 7120–7127.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. pages 5998–6008.
- 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.
- Carl Vondrick, Deniz Oktay, Hamed Pirsiavash, and Antonio Torralba. 2016. Predicting

- motivations of actions by leveraging text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 2997–3005.
- Terry Winograd. 1972. Understanding natural language. *Cognitive Psychology* 3:1–191.
- Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. **TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, pages 917–929. <https://doi.org/10.18653/v1/2020.emnlp-main.66>.
- Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*. Springer, pages 451–466.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. PMLR, pages 2048–2057.
- Tianhao Yang, Zheng-Jun Zha, and Hanwang Zhang. 2019. Making history matter: History-advantage sequence training for visual dialog. In *Proceedings of the International Conference on Computer Vision (ICCV)*.
- Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 21–29.
- Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*. pages 684–699.
- Dani Yogatama, Cyprien de Masson d’Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373* .

- 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.
- Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3712–3722.
- Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 6720–6731.
- Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. [SWAG: A large-scale adversarial dataset for grounded commonsense inference](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 93–104. <https://doi.org/10.18653/v1/D18-1009>.
- Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *ICML*.
- Jiaping Zhang, Tiancheng Zhao, and Zhou Yu. 2018a. [Multimodal hierarchical reinforcement learning policy for task-oriented visual dialog](#). In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*. pages 140–150. <https://www.aclweb.org/anthology/W18-5015>.
- Junjie Zhang, Qi Wu, Chunhua Shen, Jian Zhang, Jianfeng Lu, and Anton van den Hengel. 2018b. Goal-oriented visual question generation via intermediate rewards. In *Proceedings of the European Conference of Computer Vision (ECCV)*. pages 186–201.
- Z. Zhang, J. Singh, U. Gadiraju, and A. Anand. 2019. Dissonance between human and machine understanding. *Proceedings of the ACM on Human-Computer Interaction* 3 CSCW:1–23.

- Rui Zhao and Volker Tresp. 2018. Improving goal-oriented visual dialog agents via advanced recurrent nets with tempered policy gradient. In *Proceedings of IJCAI*.
- Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- 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.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015a. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*. pages 19–27.
- Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015b. [Aligning books and movies: Towards story-like visual explanations by watching movies and reading books](#). In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, pages 19–27. <https://doi.org/10.1109/ICCV.2015.11>.