

Cognitively-Inspired Deep Learning Approaches for Grounded Language Learning

by

Ozan Arkan Can

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of
Doctor of Philosophy

in

Computer Science and Engineering



KOÇ ÜNİVERSİTESİ

April, 2021

Cognitively-Inspired Deep Learning Approaches for Grounded Language Learning

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Ozan Arkan Can

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Deniz Yuret (Advisor)

Prof. Yücel Yemez

Assoc. Prof. Tilbe Göksun

Assoc. Prof. Erkut Erdem

Assoc. Prof. Arzucan Özgür Türkmen

Date: _____



To my love, Başak

ABSTRACT

Cognitively-Inspired Deep Learning Approaches for Grounded Language Learning

Ozan Arkan Can

Doctor of Philosophy in Computer Science and Engineering

April, 2021

Designing machines that can perceive the surrounding world and interacting with us using human language is one of the long-standing goals of artificial intelligence. Although tremendous progress has been made to model the linguistic meanings computationally, how to best integrate linguistic and perceptual processing in multi-modal tasks is a significant open problem. This thesis explores several cognitively-inspired neural architectures that consider the different aspects of the language’s role in cognition, visual perception, and task execution. Proposed models incorporate design choices motivated by cognitive science studies and are based on the common patterns in vision-language tasks.

We begin by presenting an encoder-decoder network with a novel channel-based perceptual attention mechanism and its application to the *navigational instruction following* task. The perceptual processing component of this architecture is designed to focus on individual objects and properties within the environment using the language priors while preserving the spatial relations. To benefit from the designed component, we also propose an improved agent-centric world representation to allow the model to reason over the perception spatially.

Next, we explore the usage of the Neural Module Networks approach in a real robotic system for the first time. Since collecting large-scale real world data is a labor-intensive and expensive work, the system learns the language grounding on simulated data and the perceptual representation separately to overcome the scarce data problem. However, because of the separate learning processes, inconsistencies arise between the user’s and robot’s world models. To overcome this, we propose a Bayesian learning approach that uses the implicit information in the instruction to update the perceptual belief to align what the user sees and what the robot perceives.

In both parts, we demonstrate systems that use the high-level effect of language on visual processing, which operates on high-level representations. In addition to this, in

the last part, we investigate the effect of language on low-level visual processing. To this end, we condition one or both low-level and high-level visual processing branches of a backbone architecture on language using language filters and apply these models to the *image segmentation from referring expression* task. Experiments show that modulating both low-level and high-level visual processing with language significantly improves the language grounding performance.



ÖZETÇE

Temellendirilmiş Dil Öğrenimi için Bilişsel Esinli Derin Öğrenme Yaklaşımları

Ozan Arkan Can

Bilgisayar Bilimleri ve Mühendisliği, Doktora

Nisan 2021

Bizi çevreleyen dünyayı algılayabilen ve insan dilini kullanarak bizimle etkileşim kurabilen makineler tasarlamak, yapay zekanın köklü hedeflerinden biridir. Dilbilimsel anlamları hesaplamalı olarak modellemek için önemli ilerleme kaydedilmiş olsa da, dilsel ve algısal işlemenin çok modlu görevlerde en iyi nasıl birleştirileceği önemli bir problemdir. Bu tez, dilin biliş, görsel algı ve görev yürütmedeki rolünün farklı yönlerini göz önünde bulunduran, bilişsel esinli birkaç yapay sinir ağı mimarisini sunmaktadır. Önerilen modeller, bilişsel bilim çalışmalarından esinlenilmiş ve görü-dil problemlerindeki ortak örüntülere dayanan tasarım kararlarını içermektedir.

İlk olarak, özgün kanal tabanlı bir ilgi mekanizmasına sahip bir kodlayıcı-çözücü mimarisi ve bu mimarinin *gezinim komutlarını anlama* problemine uygulanışı sunulmuştur. Bu mimarinin algı işleyen birimi, dil önceliklerini kullanarak uzamsal ilişkileri kaybetmeden ortamdaki nesne ve özelliklere odaklanacak şekilde tasarlanmıştır. Ayrıca, modelin algı üzerinde uzamsal olarak muhakeme gerçekleştirmesine olanak sağlayan gelişmiş bir dünya temsili geliştirilmiştir.

Daha sonraki bölümde, ilk kez gerçek bir robotik sistemde Sinir Modül Ağları yaklaşımının nasıl kullanılabileceği gösterilmiştir. Büyük boyutta gerçek dünya verisini toplamak maliyetli olduğundan dolayı, sınırlı veri problemine takılmamak için sistem modülleri simüle edilmiş bir veri üzerinde öğrenirken dünya temsilini ayrı olarak öğrenmektedir. Fakat bu durum, kullanıcının dünya modeli ile robotun dünya modeli arasında uyumsuzluklara sebep olmaktadır. Bu problemin üstesinden gelmek için, kullanıcının gördüğü dünya modeli ile robotun algıladığı dünya modelini eşleştirebilmek için, komutlardaki örtük bilgileri kullanarak algı temsilini güncelleyen bir Bayesçi öğrenme yaklaşımı sunulmuştur.

Her iki bölümde de, yüksek seviyeli algı temsilleri üzerinde çalışan ve dilin yüksek seviyeli görsel işleme üzerindeki etkisini kullanan sistemler gösterilmiştir. Buna ek olarak, son bölümde, dilin düşük seviyeli görsel işleme üzerindeki etkisi irdelenmiştir. Bu amaçla,

hem yüksek seviyeli hem de düşük seviyeli görsel işleme dalları olan bir mimari temel alınmıştır. Bu mimarinin bir veya her iki dalının dil filtreleri ile koşullandırılmış versiyonları *atıf ifadelerinden görüntü bölütleme* problemine uygulanmıştır. Deneyler sonucunda hem düşük seviyeli hem de yüksek seviyeli görsel işleme süreçlerini dil ile modüle etmenin, dil temellendirme performansını önemli ölçüde geliştirdiği gösterilmiştir.



ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my advisor Prof. Dr. Deniz Yuret for his immense knowledge, guidance, support and teaching me to ask important questions. I am thankful to him for introducing me the Grounded Language Learning field and encouraging me to work on the challenging problems. I would like to thank Assoc. Prof. Tilbe Göksun for sharing her knowledge and insights about how we learn and use language. I am also grateful to Prof. Yücel Yemez, Assoc. Prof. Arzucan Özgür Türkmen and Assoc. Prof. Erkut Erdem for participating in my thesis committee. I am also thankful to Assoc.Prof.Dr. Gülşen Cebiroğlu Eryiğit for introducing me the academic world during my undergrad studies.

I would also thank my friends in AI Lab. Among them, I would like to especially thank Onur Kuru, Osman Başkaya, İlker Kesen, Erenay Dayanık and Ömer Kırnap. We have learned a lot from each other, shared fantastic ideas and consumed too much coffee together. Their friendship made the AI Lab experience more enjoyable.

I acknowledge the financial support for my doctoral studies provided by Koç University, Technological Research Council of Turkey (TUBITAK).

Last but not the least, I want to thank my family for the value they showed for my education. I especially thank my mom for exceeding her limits to provide me an eye-opening life experience. Finally, a huge thanks to the love of my life, Başak, who has been with me from the starting of this long journey. She has been the one who raises me again from all my downs. Without her unconditional love and support, I am certain that this thesis wouldn't become a reality.

TABLE OF CONTENTS

List of Tables	xii
List of Figures	xiv
Abbreviations	xvii
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Background	3
1.2.1 Rule-based Approaches	4
1.2.2 Semantic Parsing	5
1.2.3 Neural Models	6
1.3 Thesis Contributions and Overview	7
Chapter 2: Learning to Understand Navigational Instructions	11
2.1 Introduction	11
2.2 Model	15
2.2.1 Input Representation	15
2.2.2 The Neural Architecture	17
2.2.3 Training	21
2.2.4 Inference	22
2.3 SAIL Dataset	22
2.4 Synthetic Dataset (SAILx)	24
2.4.1 Map and Path Generation	25
2.4.2 Instruction Generation	25
2.4.3 Instruction Coverage	25

2.4.4	The Fixed-Sized SAILx Dataset	26
2.5	Experiments & Results	26
2.5.1	Baselines	27
2.5.2	SAIL dataset	27
2.5.3	SAILx dataset	30
2.5.4	Visualization of Attention Weights	33
2.6	Related Work	34
2.7	Conclusions	36
Chapter 3:	Learning from Implicit Information in Natural Language In-	
	structions for Robotic Manipulations	37
3.1	Introduction	37
3.2	Preliminaries	40
3.2.1	Visual Object Anchoring	40
3.2.2	Natural Language Grounding	42
3.3	System Description	45
3.4	Resolving Inconsistencies	47
3.5	Experiments	49
3.5.1	Synthetic Data	49
3.5.2	Training	50
3.6	Showcase	52
3.7	Related Work	52
3.8	Conclusions	55
Chapter 4:	Language Controls More Than Top-Down Attention: Modulat-	
	ing Bottom-Up Visual Processing with Referring Expressions	57
4.1	Introduction	57
4.2	Model	59
4.2.1	Low-level Image Features	59

4.2.2	Language Representation	60
4.2.3	Segmentation Model	61
4.2.4	Learning	63
4.3	Experiments	63
4.3.1	Datasets and Experiment Setup	63
4.3.2	Ablation Results	65
4.3.3	Quantitative Results	67
4.3.4	Qualitative Results	68
4.4	Related Work	72
4.4.1	Semantic Segmentation	72
4.4.2	Referring Expression Comprehension	72
4.4.3	Image Segmentation from Referring Expressions	73
4.4.4	Language-Conditional Filters	74
4.5	Conclusions	75
Chapter 5:	Conclusion & Future Directions	76
5.1	Future Directions	77
5.1.1	Learning to Simulate the World	77
5.1.2	Language as Reward	79
5.1.3	Grounded Language Learning in the Social World	80
	Bibliography	81
Appendix A:	Learning to Understand Navigational Instructions	111

LIST OF TABLES

2.1	Instruction types in SAIL with their frequencies, definitions and examples.	23
2.2	The instruction types and coverage statistics for our generator. The second column shows the frequency of each task in the SAIL dataset. The third column gives the percentage of the original instructions SAILx is able to generate. The last column gives the percentage SAILx can generate for instructions that occur more than once.	26
2.3	Task distributions in the fixed-sized SAILx data.	27
2.4	Overall results. Results of the previous works were obtained using the “vTest” procedure if not explicitly stated otherwise. † denotes the vDev protocol. ‡ states that ensembling is used for the inference. The numbers in the parentheses are the standard deviations.	28
2.5	The performance (test accuracy) comparison of baselines and the proposed model. Results were obtained using ensemble of ten models for each architecture. Experiments have been done using the schema proposed in this study.	30
2.6	Results on the generated fixed data. The average and ensemble scores were obtained by using 10 models for each architecture. The numbers in the parentheses are the standard deviations.	32
4.1	Ablation results and comparison with the previous works on the val set of UNC dataset with $prec@X$ and IoU metrics.	65

4.2	Comparison with the previous works on four datasets. Evaluation metric is the overall <i>IoU</i> and higher is better. Bold scores indicate the state-of-the-art performances. “-” indicates that the model has not been evaluated on the dataset. “n/a” indicates that splits are not same.	67
A.1	Example instructions generated for the segments in Figure A.1.	113
A.2	Example text templates from the data generator.	114
A.3	Hyperparameters used in experiments.	115



LIST OF FIGURES

2.1	An illustration of a map and a set of instructions from the SAIL dataset [MacMahon et al., 2006]. The letters indicate items, the thin rectangles indicate wall paintings, and the thick rectangles indicate floor patterns. The circle represents the initial position of the agent and the red arrow shows the initial direction. Aqua arrows represent the execution of the instruction set: “Take the pink path to the red brick intersection. Go right on red brick. At the wood intersection, turn left. Position one is where the sofa is.”	12
2.2	Spatial relations between items: The circle represents the agent. The following narrative describes that the agent is supposed to stay at the current position: ”you should be two alleys away from a sofa and then a hatrack beyond that”. To process the given instruction, the agent should understand the distance between itself and the items in the instruction, and their relative ordering.	16
2.3	An agent-centric grid representation for the perceptual information. . . .	17
2.4	The convolutional neural network architecture with channel attention. The perceptual input p_t is a 5×20 grid and c is the length of the binary vector representation (Section 2.2.1) for each grid position. The first layer applies a $(1, m, c, d_1)$ filter bank, where $1 \times m$ is the size of the filter and d_1 is the number of filters. The output of the first layer is a $(5, n, d_1)$ tensor where $n = 20 - m + 1$. β_t is the attention vector determined by the decoder hidden state. After further convolutional layers, the output of the whole architecture, c_t , has (k, l, d_2) dimensions.	19
2.5	The distribution of the length of action sequences.	24

2.6	The learning efficiency comparison between the baseline models and the grid based model. The y axis shows the number of instances required to reach 90% accuracy on unseen data.	31
2.7	t-SNE visualization of attention weights.	33
3.1	The semiotic cycle [Steels and Hild, 2012] shows the communication process between the speaker and the hearer.	37
3.2	Demonstration of the language grounding process for the instruction “pick up the apple to the right of the black mug”. Anchoring system sends the snapshot of the anchors (1). Then, a preprocessor transforms the anchors into a grid representation which the language grounding system operates on (2). The parser parses the given instruction and generates a computation graph which specifies the execution order of neural modules (3). Finally, the neural modules are executed according to the computation graph to produce the action (4).	43
3.3	A depiction of both used physical system setup (upper), as well as used software architecture (lower). The arrows represent the flow of data between the modules of the software architecture. Blue solid arrows and boxes illustrate the preliminary system (outlined in Section 3.2), while red dashed arrows and boxes illustrate the novel extension for reasoning about different symbolic label configurations (and hence resolving inconsistencies between language and perception), by using Bayesian learning (as presented in Section 3.4).	46
3.4	Learning curve of the neural modules.	51

3.5	We give the robot the instruction: “ <i>pick up the ball in front of the can</i> ”. The robot executes the action and waits for further instructions. We then give the instruction to “ <i>drop it in front of the mug</i> ”. The problem in this step is that there is no object classified as ‘mug’, which means that none of the objects has as label with highest probability ‘mug’. We correct for this through probabilistic reasoning over not only the top label for each object but a number of top ranked labels per object. This allows the anchoring system to correct its classification of an object based on what we as humans think an object is. Given the instruction, the anchoring system re-classifies the black object from ‘pot’ to ‘mug’. The instruction is then successfully carried out. The recorded video can be found here: https://vimeo.com/302072685	53
4.1	Overview of our model.	60
4.2	Visualization of some of the correct predictions of our model. First column shows the original images and others show the output of the model for the given referring expressions. Examples are taken from the UNC validation set.	69
4.3	Incremental segmentation result of our model on UNC test split instance.	70
4.4	Visualization of some of the incorrect predictions of our model. Each group (a-d) shows one of the pattern we observed within the predictions. In each group, the first column shows the original image, the second one is the ground truth mask and the third one is the prediction of our model. Examples are taken from the UNC validation set.	71
A.1	An example map and two segmented paths generated by our artificial data generation framework.	112

ABBREVIATIONS

DL	Deep Learning
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
NMN	Neural Module Networks

Chapter 1

INTRODUCTION

1.1 Motivation

As human beings, we can easily understand the surrounding environment with our perceptual system and interact with each other using language. Consider the expression “the sandwich next to the white bowl”. When we hear this expression, we are able to relate each word to the actual objects (e.g., associating the *sandwich* word to the actual sandwich instance) or relations (e.g., associating the *next to* preposition to the spatial relation between the sandwich object and the white bowl object) in the surrounding environment. We are also capable of understanding each other’s directives such as “take the pink path to the red brick intersection” or “pick up the ball in front of the can”. Using our intelligence, we perceive, understand, and jointly reason over the natural language and the visual world surrounding us.

Similarly, an intelligent agent should understand both perceptual and linguistic inputs, and reason over both modalities. A visual perception system should be able to localize objects referred with natural language inputs in addition to recognizing objects’ classes and their attributes, such as “a red vase that is to the right of two other similar red vases”. To follow an instruction given in natural language such as “go until the wood intersection”, an intelligent robot should map the words or phrases to the visual entities (e.g. *wood intersection*) in the environment and physical actions (e.g. *go until*).

Designing machines that can understand human language is one of the long-standing goals of artificial intelligence. Although tremendous progress has been made to model the linguistic meanings computationally, in the symbolic AI studies, semantics is usually represented with symbolic logic or other words [Miller, 1995, Saint-Dizier and Viegas,

1995, Liu and Singh, 2004]. In contrast to this, a child starts to learn his mother language by connecting words with physical objects and events by observing the co-occurrence of linguistic and perceptual information [Saffran et al., 1999, Saffran, 2003] in a spatial and social world [Smith and Gasser, 2005, Samuelson et al., 2011, Slone et al., 2019], and further improves language abilities using social and linguistic cues [Hollich et al., 2000, Baldwin and Moses, 2001, Golinkoff and Hirsh-Pasek, 2006, Nelson, 2007, Brooks and Meltzoff, 2008, Tomasello, 2010, Parish-Morris et al., 2013, Hawkins et al., 2018]. Correspondingly, language grounding studies aim to capture the linguistic meaning by mapping the words to the relevant aspects of the non-linguistic world. Systems that link the language and world may be called *situated* to state the relation between the linguistic and dynamically changing non-linguistic world. These systems may also be termed as *grounded* due to addressing the symbol grounding problem [Harnad, 1990]. This domain includes systems that translate language into physical actions, systems that make linguistic decisions using non-linguistic input, and systems that learn how to use and generate the language in a grounded setting. These systems interact with the non-linguistic world via gathering information from sensors and/or knowledge bases or performing actions with actuators or queries.

A system that interconnects the language and the physical world should contain some vital components working in harmony to perform several subtasks. These subtasks include processing individual modalities, creating links between them, reasoning over the connection, performing actions in the physical world, or maintaining a memory for the history of language and world. In this manner, Deep Learning (DL) would be a suitable approach to perform in this domain due to the flexibility at defining architectures that consist of various components that each processes a modality, and the ability to train them as a whole. Recent success of neural models on vision [Krizhevsky et al., 2012, Mnih et al., 2014, He et al., 2016] and language [Mikolov et al., 2010a, Sutskever et al., 2014] tasks has brought the attention of both computer vision and natural language processing communities to the common tasks; image captioning [Vinyals et al., 2015b, Xu et al., 2015], visual question answering [Antol et al., 2015, Malinowski et al., 2015, Andreas et al., 2016a], and visual dialog [Das et al., 2016]. However, how to best integrate linguistic

and perceptual processing is still an important open problem.

In this thesis, we explore different cognitively-inspired neural architectures that consider the different aspects of the language’s role in cognition, visual perception and task execution. Proposed models incorporate design choices motivated by the cognitive science studies and based on the common patterns in vision-language tasks. We show that building neural architectures that are exploiting the injected inductive biases can lead to better accuracy and generalization on various vision-language tasks, including *navigational instruction following*, *robotics instruction understanding*, and *image segmentation from referring expressions*.

1.2 Background

Early cognitive science studies assumed that the language processing is a high-level cognitive function and engages with the amodal, symbolic conceptual knowledge, and there is no link between the sensorimotor experience and linguistic processing [Fodor, 1975, Kintsch and Van Dijk, 1978, Fodor, 1983, Jackendoff, 2002]. In contrast to this, based on the empirical findings, grounded cognition claims that knowledge is represented with the modal representations, and those representations are built from the embodied experiences and actions taken in the world [Proffitt, 2006, Barsalou, 2008]. This view is intriguing because it proposes a solution to the symbol grounding problem [Harnad, 1990] by connecting the high-level symbolic information to the mental representations built from the low-level sensorimotor information experienced in the real world [Varela et al., 2016].

This thesis extends the years of work in artificial intelligence about the symbol grounding problem. First, [Searle, 1980] demonstrated the symbol grounding problem with the famous *Chinese Room* metaphor and challenged the core idea of symbolic AI. According to this idea, a computer machine passes the Turing Test [Turing, 1950] in Chinese by responding to the Chinese string inputs with Chinese string outputs that are distinguishable from a Chinese speaker’s responses, then the computer would understand the meaning of Chinese symbols as we understand the English symbols. Searle’s demonstration was simply replacing himself with the computer, receiving Chinese symbol inputs, manipulating them with the rules depending on their shape, and finally returning a response in

Chinese symbols. Since Searle does not speak Chinese, he would not understand Chinese under those circumstances. Hence, neither the computer could understand Chinese. Taking from this, [Harnad, 1990] described the “*symbol grounding problem*” in symbolic AI as grounding the meaning of symbols to the other meaningless symbols, and proposed that the elementary symbols should ground to the sensory information. Similarly, [Coradeschi and Saffiotti, 2003] proposed the definition *anchoring* as “*the process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical object*” for the robotics domain. In addition to grounding the symbols to the perceptual inputs, [Ziemke et al., 1999] discussed the need for *embodiment* for symbol grounding.

There are many studies that are in line with the grounded learning approach. We can categorize these approaches as rule-based approaches, grammar-based semantic parsing approaches, and more recently neural network approaches.

1.2.1 Rule-based Approaches

Winograd’s SHRDLU put the grounded language learning in the spotlight by manually integrating the visual world with language in the blocks domain [Winograd, 1973]. [Nilsson, 1984] connected the perception with the symbolic representation in the robot Shakey, for the first time. More recently, [Siskind, 1993, Siskind, 1995, Siskind, 2001] built a system based on force-dynamics relations between objects observed visually, and provided basis for the semantic roles of the participants for verbs. [Narayanan, 1999] developed a representation for action verbs that are used to control a simulated robotic arm. This representation is also used as a basis for understanding physically grounding metaphors. [Roy, 2005] showed that the symbolic modeling is inadequate to capture the context dependent word meaning and proposed to use the first-person perspective sensory data in grounded learning. [Goertzel et al., 2010] developed a software system where a virtual agent is able to carry out simple language instructions grounded in its perceptions and actions in a virtual environment. This system uses knowledge bases such FrameNet [Baker et al., 1998] to associate semantic meaning to linguistic utterances. [Mavridis and Roy, 2005] have proposed the idea of the *grounded situation model* which merges the different

sensing modalities including the verbal one. Their framework also manages the interaction history. They implemented the framework on a simple setup (a robotic arm and colored balls). Although these systems show some grounded language understanding capabilities, they are consist of hand-designed components that are not learned from experiences or data, which limits their language understanding capabilities.

1.2.2 Semantic Parsing

Most studied approach in this domain is mapping natural language instructions into a formal meaning representation. $\lambda a.move(a) \wedge dir(a, forward) \wedge len(a, 2) \wedge to(a, \iota x.chair(x))$ is a lambda calculus formal representation that corresponds to the instruction, "move forward twice to the chair". *move*, *dir*, *len*, *to* are predicates and take some arguments that represent some objects or properties in the environment such as an agent (*a*) and a direction (*forward*). This formal representation should be executable in order to be used for instruction following. Hence, to use this methodology one should build an ontology consist of a set of predicates. This ontology could be learned or hand-crafted.

[Matuszek et al., 2010, Matuszek et al., 2012] proposed supervised algorithms to train semantic parsers for navigational instruction following. [Chen and Mooney, 2011] presented a system that translates instructions to formal executable plans by training KRISP semantic parser [Kate and Mooney, 2006] with the aligned instruction and action sequences pairs. Later, [Chen, 2012] improved the system by modifying the underlying semantic grammar and tested the system also on Chinese instructions. [Kim and Mooney, 2012] adapted the grounded language learning as the probabilistic context free grammar (PCFG) induction introduced by [Börschinger et al., 2011] to the navigation domain. They further improve this technique using a re-ranking module with the task-specific weak signal [Kim and Mooney, 2013].

One of the widely used grammars in semantic parsers is Combinatory Categorical Grammar (CCG) [Steedman, 2000]. [Zettlemoyer and Collins, 2005, Zettlemoyer and Collins, 2007] proposed supervised learning algorithms for CCG grammar. Artzi and Zettlemoyer [Artzi and Zettlemoyer, 2013] learned a semantic parser seeded with a manual lexicon. The parser operates on a combinatory categorical grammar (CCG) to translate

the instruction into a lambda-calculus formalism for the semantic representation. They improved their system using a re-ranker and controlling the size of the lexicon by a data driven fashion [Artzi et al., 2014].

Although the semantic parsers, that operate on the formal meaning representations, can handle complex enough instructions, it is an expensive engineering challenge to design and learn the executable ontology. In general, this limits the scalability of the use of semantic parsers.

1.2.3 Neural Models

With the recent success of deep learning [LeCun et al., 2015], the research direction in grounded language learning and understanding has started to move towards the neural models from rule-based and grammar-based approaches. In particular, recurrent neural networks (RNN) has shown great success on language modelling [Mikolov et al., 2010b], syntactic parsing [Vinyals et al., 2015a], and machine translation [Sutskever et al., 2014]. Similarly, convolutional neural networks (CNN) has outperformed previous studies on image classification [Krizhevsky et al., 2012, Simonyan and Zisserman, 2014, He et al., 2016], object recognition [Ren et al., 2015] and semantic segmentation [Long et al., 2015, Ronneberger et al., 2015]. With the success of both architectures, it became a paradigm to process the language inputs (such as expressions, questions, and instructions) with recurrent neural networks, to process the perceptual inputs (such as images, videos, representation of the agent’s view and state representations) with the convolutional neural networks, and then fusing the results of both parts to generate outputs (such as answers in question answering, image segments or object locations in grounding referring expressions and action sequences in instruction following). The aforementioned RNN-CNN approaches have been widely used in language-vision tasks such as image segmentation from referring expressions [Hu et al., 2016b], image captioning [Vinyals et al., 2015b], visual question answering [Antol et al., 2015] and navigational instruction following [Anderson et al., 2018b].

To fuse both linguistic and visual representations, different techniques have been used such as concatenation [Malinowski et al., 2015] and element-wise multiplication

[Malinowski et al., 2015, Lu et al., 2016, Kim et al., 2016]. However, the widely used method is to utilize the attention mechanism [Bahdanau et al., 2014] to focus on different parts of inputs while generating an output. [Xu et al., 2015] utilize the attention mechanism in image-captioning task, where the model learns to slide the attention on the image while generating the corresponding words. [Shih et al., 2016, Xiong et al., 2016, Yang et al., 2016] have explored attention based models for visual question answering. Similar to these systems, we use the attention mechanism in the proposed models to highlight the important parts of the inputs to relate language with the non-linguistic input.

One neural network family distinguishing from the end-to-end network is Neural Module Networks [Andreas et al., 2016b]. In this approach a parser converts the language into a computational program. Applying this program onto the visual representation grounds the language and generates the result. The computational program consists of the learnable neural modules. [Andreas et al., 2016a] extended this approach by learning to select the correct graph. [Johnson et al., 2017b] and [Hu et al., 2017a] developed reinforcement learning based approaches to learn to generate the computational graph. In this thesis, we explore both end-to-end networks and the neural module networks approach for grounded language learning.

1.3 Thesis Contributions and Overview

In this thesis, we present cognitively-inspired neural architectures for grounding language onto an embodied agent’s perceptual states to take decisions to follow instructions given in natural language, the intermediate world representations obtained from the perceptual system of a table-top robot to execute robotic instructions, and images to segment objects referred with natural language.

Chapter 2 presents an end-to-end architecture with a channel-based perceptual attention mechanism for navigational instruction following task. In this task, an agent stays in a maze-like world and gets the sensory information from the line of sight. The agent is asked to navigate from a position to the target position, which is described with a free-form natural language. The aim of the agent is to follow the instruction by generating a sequence of actions. Consider the instruction, “at the chair, turn right”. In this example, language

controls the perceptual focus by highlighting the chair object. Prior works [Mei et al., 2016, Kočiský et al., 2016, Fried et al., 2018, Anderson et al., 2018b], mostly focus on the words by applying attention mechanism onto the instruction to select the critical word at each time step while generating action sequence. These studies encode the perceptual input with a binary representation and process this representation as a whole by not focusing the individual objects. However, in our cognition system, comprehending words activates the visual properties corresponding to the concepts referred with those words [Simmons et al., 2007, Evans, 2009, Pulvermüller, 2013] and sets high-level priors on the visual system to recognize those concepts [Lupyan and Clark, 2015].

Motivated by the idea of words as high-level priors on the visual system, we present an end-to-end architecture that dynamically controls the focus on the perceptual input and reason over it spatially while generating an action. The designed perception component of the model first applies a novel channel-wise attention to focus on individual objects and properties within the perceptual input without losing the spatial relations. Then, the rest of the component processes the modulated input to take into account the high-level spatial relations. To actually benefit from this ability, we propose an agent-centric grid representation that preserves the spatial dimensions in the perceptual environment instead of using a multi-hot vector representation that loses the spatial relations between objects. We show the effectiveness of our model on the available SAIL benchmark dataset [MacMahon et al., 2006]. However, our detailed analysis shows that the SAIL dataset has certain problems regarding its size and balance of instruction types to truly differentiate the models. To overcome the sparse and unbalanced data problems, we propose SAILx, a synthetic data generator. Using the unlimited data generated by SAILx, we show that our proposed model achieves better generalization with less data than the benchmark models.

In **Chapter 3**, we present a system for language grounding in a real robot. To execute an instruction such as “pick up the black bowl to the right of the small plate”, the robot should ground this instruction onto its perceptual world representation, which should be aligned with what the speaker sees. Due to the probabilistic nature of the perceptual system, contradictions arise between the world representations of a human and a robot. To achieve an alignment between the speaker’s (a human user) and the hearer’s (a robot)

mental world models [Stolk et al., 2016], we present a Bayesian learning approach that exploits spatio-relational information that is implicitly present in instructions given by a human. Recent studies suggest that language could improve the human visual system for the detection of objects otherwise salient to the visual system [Lupyan and Ward, 2013, Boutonnet et al., 2013, Lupyan and Clark, 2015] or manipulate how we interpret a visual stimuli [Dils and Boroditsky, 2010, Dumitru et al., 2013]. Similarly, we highlight the object labels consistent with the grounding by assuming that the objects and their attributes referred to in the instructions are aligned with the configuration in the environment.

Ideally, the visual perception system and the language grounding system would be learned jointly [Hu et al., 2017a, Johnson et al., 2017b, Santoro et al., 2017, Hudson and Manning, 2018, Perez et al., 2018]. However, due to the data scarcity problem in the robotics domain, we opt to learn the visual perception system and the language grounding component separately. We use the perceptual anchoring [Coradeschi and Saffiotti, 2000] as a world representation and map the resulted anchors to a grid world representation similar to **Chapter 2**. We use the Neural Module Networks [Andreas et al., 2016b] approach for the first time for language grounding in a real robot. We show the generalization and compositionality of designed neural components with a simulated *Naming Game* [Steels, 1995]. We demonstrate the validity of our approach for reconciling instructions and world representations on a showcase scenario involving a camera, a robot arm, and a natural language interface.

The language grounding systems proposed in Chapter 2 and Chapter 3 demonstrate the *high-level* effect of language on perception. In both studies, the designed models operate on the high-level perceptual representation, and language drives the “*top-down*” visual attention. However, recent studies from cognitive science point out that language comprehension also affects low-level visual processing [Meteyard et al., 2007, Boutonnet and Lupyan, 2015]. Motivated by this, we propose a model that can modulate either or both of “*bottom-up*” and “*top-down*” visual processing with language. Recent deep learning system for language grounding typically start with an image processing component which is not conditioned on language, then connect the language representation with *high-level* visual features [Malinowski et al., 2015, Xu et al., 2015, Lu et al., 2016, Kim

et al., 2016, Anderson et al., 2018a, Zellers et al., 2019]. Our model differs from these architectures by having explicit “*bottom-up*” and “*top-down*” visual processing branches which are conditioned on language.

We evaluate our proposed model on the task of *image segmentation from referring expressions* where given an image and a natural language description, the model returns a segmentation mask that marks the object(s) described. Experiments show that modulating both “*bottom-up*” and “*top-down*” visual processing with language improves the language grounding performance by a large margin. With this contribution, our proposed model achieved state-of-the-art or comparable performance on four different referring expression datasets.

Finally, in **Chapter 5**, we summarize the contributions provided by this thesis and discuss the remaining challenges along with the potential research directions.

Chapter 2

LEARNING TO UNDERSTAND NAVIGATIONAL INSTRUCTIONS

2.1 Introduction

In the *navigational instruction following* task (Figure 2.1), an agent in a maze like environment receives sensory information from its line of sight. The agent is asked to navigate from a starting position to a target position using free-form natural language instructions. The aim of the agent is to follow the instructions by generating a sequence of actions. The agent can take one of four possible actions, $\{MOVE, RIGHT, LEFT, STOP\}$. The *RIGHT* and *LEFT* change the orientation of the agent, whereas the *MOVE* transports the agent to the next position in the direction it is facing. The agent ends its trip if it takes the *STOP* action, hits a wall, or exceeds the maximum number of actions.

Consider the instructions in Figure 2.1. To follow such instructions, an agent should resolve the references to objects (e.g., "pink path", "the sofa"), understand the actions (e.g., "turn right", "take the ..."), understand implicit requests (e.g., "at the wood intersection"), and reason over the environment to understand goal positions. Here, language controls the visual attention by highlighting the objects and properties in the environment. The agent should consider these highlights while taking the actions.

Recent cognitive science studies suggest that hearing words that refer the concepts in the environment activates the corresponding brain regions for visual representation [Simmons et al., 2007, Evans, 2009, Pulvermüller, 2013] and helps to recognize the correspondent in the subsequent visual inputs [Lupyan and Clark, 2015]. Even, [Lupyan and Swingley, 2012] showed that in addition to thinking about the object, saying the label of the object improves the accuracy of visual search. Similarly, [Richardson et al., 2001, Richardson et al., 2003] demonstrated that comprehension of verbs implying spatial directional paths

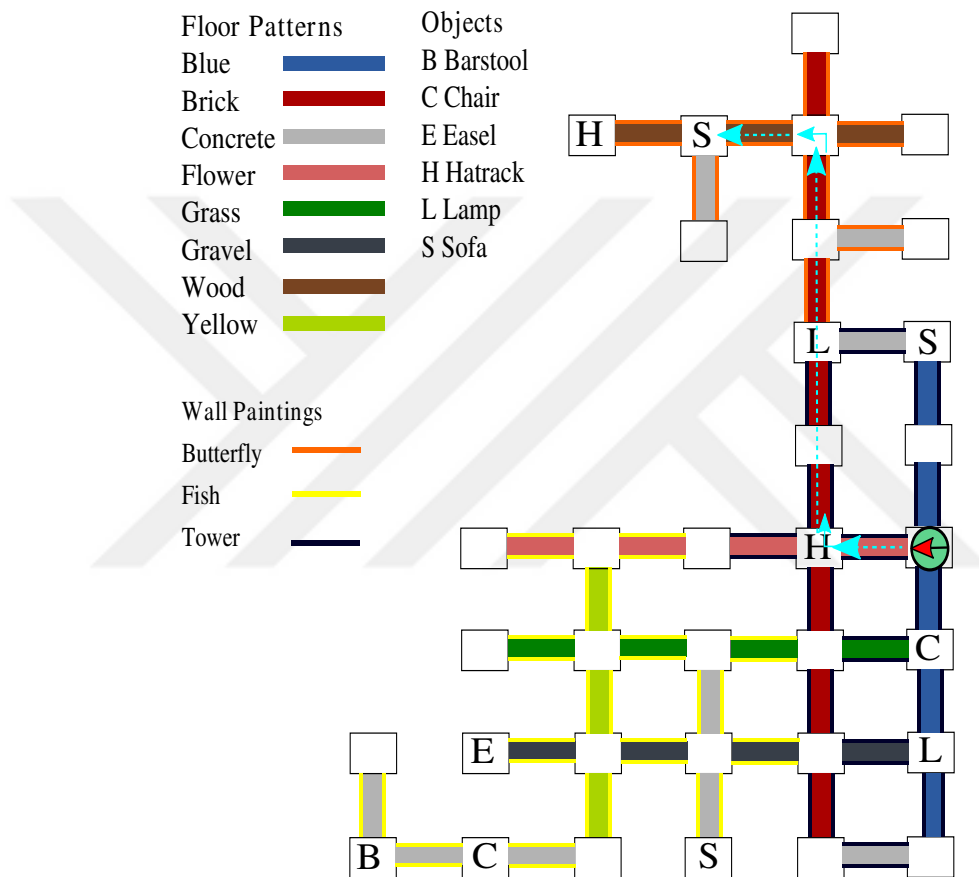


Figure 2.1: An illustration of a map and a set of instructions from the SAIL dataset [MacMahon et al., 2006]. The letters indicate items, the thin rectangles indicate wall paintings, and the thick rectangles indicate floor patterns. The circle represents the initial position of the agent and the red arrow shows the initial direction. Aqua arrows represent the execution of the instruction set: “Take the pink path to the red brick intersection. Go right on red brick. At the wood intersection, turn left. Position one is where the sofa is.”

influences the visual processing of the objects during the visual discrimination task. In line with that, empirical studies suggest that seeing a visual input that is not coherent with the described directional perspective in a sentence affects the judgement [Stanfield and Zwaan, 2001, Richardson et al., 2003, Borghi et al., 2004]. Additionally, [Choi and Bowerman, 1991, Bowerman and Levinson, , Göksun et al., 2017, Aktan-Erciyes and Göksun, 2019] discussed that language guides attention to recognize the specific spatial components of the spatial events. Combined, these results suggest that comprehending words sets predictive high-level priors on the visual system to detect objects and spatial relations in the upcoming visual input.

Motivated by this idea, we propose an encoder decoder neural network architecture with a novel channel attention mechanism, which enables the model to focus on the objects in the upcoming perceptual input without losing the spatial relations (Section 2.2.2). In contrast to this, prior works [Mei et al., 2016, Kočiský et al., 2016, Fried et al., 2018, Anderson et al., 2018b], process the perceptual input as a whole without considering the spatial relations between the objects. To model language and the world jointly, having descriptive enough representations is a crucial requirement for grounded language understanding. Previous studies [Chen and Mooney, 2011, Chen, 2012, Artzi and Zettlemoyer, 2013, Artzi et al., 2014, Mei et al., 2016, Fried et al., 2018] indicate only the existence of objects in a direction in the perceptual representation. However, this representation misses some spatial relations between objects themselves, and the agent, such as the distance between objects, the order of objects, and their relative positions with respect to the agent. To capture all the spatial information available to the agent, we propose an agent-centric grid based representation where we map the agent’s view into a grid without losing positional information (Section 2.2.1). We evaluate our model that takes advantage of the proposed perceptual representation on the standard SAIL benchmark dataset [MacMahon et al., 2006]. Our proposed model achieves state-of-the-art results.

The SAIL dataset has certain problems regarding its size and balance of instruction types. It consists of 3237 instruction sentences with associated worlds and actions. Even though the task domain and vocabulary are fairly small, this number may be insufficient for a couple of reasons. One third of the dataset is typically used as a test set, and this may not

be large enough to differentiate models close in performance with statistical significance. Certain words or tasks occur too few times in the training set to meaningfully generalize their meaning. We often observe correct actions performed for the wrong reasons in state-of-the-art models (e.g. “move to the chair” always causing one step to be taken, because that was the correct action in the only example in the training set).

Controlling the balance between different types of instructions may also be important to get a fair diagnosis of a model. Table 2.1 shows our rough categorization of the tasks found in the SAIL dataset. For each type, we state the proportion of sentences in the dataset of that type, the description of the task from the perspective of the agent, and example instructions. One striking observation is that roughly a third of the dataset consists of “language only” instructions, i.e. instructions such as “turn right” that the agent can follow without any perceptual information about the world. Our experiments show that a blind agent that does not perceive the world can guess the correct action for 64% of the instruction sentences where the state-of-the-art performance is around 70%. On the other hand, semantically more complicated categories such as “move until” may not have enough examples that will allow a learning agent to generalize correctly.

To overcome the sparse and unbalanced data problems, we propose SAILx, a synthetic data generator (Section 2.4) following the recent studies on artificial data generation. [Goyal et al., 2017] and [Agrawal et al., 2016] show that human annotated datasets may contain latent biases and neural networks are able to leverage those biases to achieve high performance. [Kuhnle and Copestake, 2018] and [Kielbaso et al., 2016] claim that artificially generated data allows us to examine language understanding abilities of multimodal systems. Following this direction, our data generation procedure provides control over the task, the language, and the world with the ability to focus on individual grounding problems.

In addition, we think the standard way of evaluating models comparing their performance on a fixed-sized dataset may not be adequate. A fixed-sized dataset that is too small may suffer from sparsity and balance problems. A fixed-sized dataset that is too large may allow many models with enough capacity to solve a given task without differentiating their sample efficiency. We recommend comparing models using the number of examples they

take to learn a given task to an acceptable level of accuracy. We evaluate our model and compare it to others by measuring the number of instances they take to reach 90% test set accuracy on various tasks with the SAILx generator (Section 2.5.3).

Our key contributions include,

- We present a state-of-the-art model with a channel based perceptual attention mechanism and improved world representation.
- We develop a synthetic data generation framework to overcome the small and unbalanced dataset problems.
- We use an experimental methodology that compares models in terms of their learning efficiency.

2.2 Model

We use an encoder decoder neural network model to learn the meanings of navigational instructions. To our knowledge ours is the first model in the navigation domain that uses an agent-centric grid based perceptual representation and a channel-based attention mechanism. We first describe input representations for the natural language instructions and perceptual states. Next, we describe the architecture of the model. Finally, we outline the training and inference procedures.

2.2.1 Input Representation

Natural Language Instructions

Since the instructors were allowed to use free-form language, the instructions contain uppercase letters, hyphenations (e.g. *blue-tiled*), shortened words (e.g. *fwd* as short version of *forward*) and typos (e.g. *aesal* instead of *easel*) in the SAIL dataset (The SAILx dataset does not contain any misspellings). Considering the small number of instructions, we split hyphenated words into subwords to prevent sparsity. We left shortened words and typos as they are. We represented each word in the vocabulary with a one-hot vector.

Perceptual States

In previous studies, a common choice to represent the perceptual information for the agent is the concatenation of bag-of-features vectors for each direction and the agent’s current position [Mei et al., 2016]. For the SAIL dataset, the agent can observe items, flooring patterns and wall paintings of the halls. Although this representation captures the features and the directions of items, it can not capture the spatial relations.

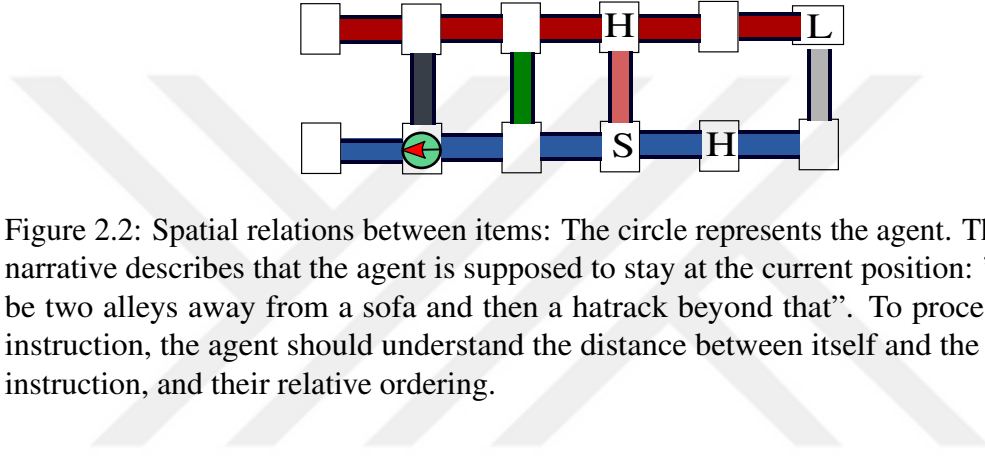


Figure 2.2: Spatial relations between items: The circle represents the agent. The following narrative describes that the agent is supposed to stay at the current position: ”you should be two alleys away from a sofa and then a hatrack beyond that”. To process the given instruction, the agent should understand the distance between itself and the items in the instruction, and their relative ordering.

Figure 2.2 illustrates the need for understanding of spatial relations. The current position of the agent is the targeted final position for a paragraph. The last instruction of that paragraph describes the final position referring to the surrounding objects. The agent must be able to recognize its position and the relative position of the surrounding objects to understand the given instruction.

To capture the spatial properties of items and their relations with the agent, we propose an agent-centric grid representation for perception (Figure 2.3). In this representation, the agent senses the world as a grid of cells in the four primary directions. Each cell contains a binary vector representing a node, a hall or a non-walkable cell. If the cell is a node, one of the item bits is set. If it is a hall the corresponding bits for the floor and wall patterns are set. In addition to material bits, one of the last three bits specifies the type of the cell.

We always fill the grid in an agent-centric orientation. Each row contains features about a direction. The first cell of each row always contains information about the current location of the agent. We fill the grid in a clockwise manner. We copy the first row to the last row to preserve the geometric relations between two consecutive directions. We use 20

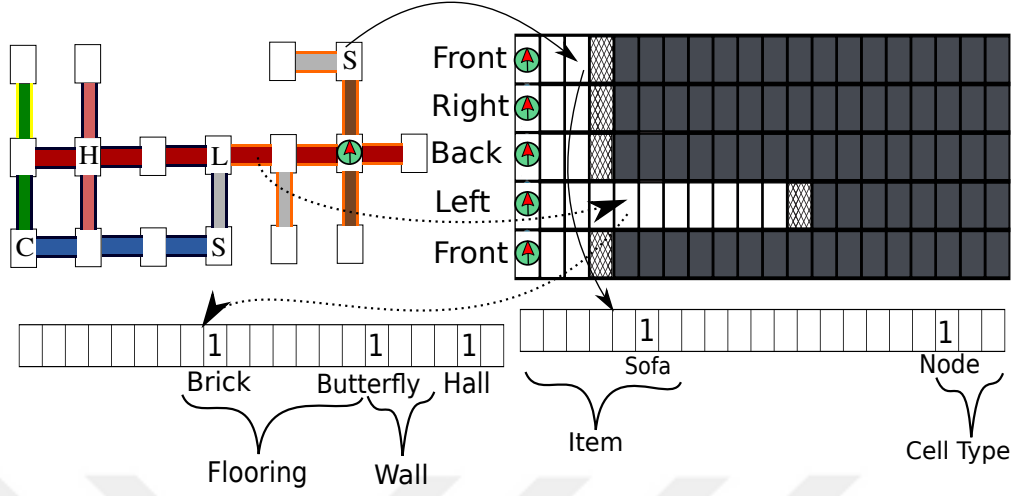


Figure 2.3: An agent-centric grid representation for the perceptual information.

columns to fit the actual maps into this grid representation.

2.2.2 The Neural Architecture

Our model uses an encoder decoder neural architecture [Sutskever et al., 2014] with a channel-based perceptual attention module. It consists of three major components: an encoder, a decoder, and a convolutional neural network (CNN) with a channel-based attention mechanism. The encoder and the decoder model the input language and output action sequences using Long Short-Term Memory (LSTM) units [Hochreiter and Schmidhuber, 1997a]. The CNN processes the perceptual input and the attention mechanism controls the focus on the features captured by the CNN. We give a detailed description of each component below.

Language Encoder

Similar to [Mei et al., 2016], the encoder takes the natural language instruction as a sequence of one-hot word vectors ($\mathbf{w} = (w_1, w_2, \dots, w_I)$) and uses a bidirectional LSTM [Graves and Schmidhuber, 2005] to process the sequence in both forward and backward

directions. It produces a hidden state for each position with the following formulation:

$$\begin{aligned}
 x_i &= W_e w_i \\
 f_i &= \text{LSTM}(W_f, x_i, f_{i-1}) \\
 b_i &= \text{LSTM}(W_b, x_i, b_{i+1}) \\
 h &= [f_I; b_1]
 \end{aligned} \tag{2.1}$$

where W_e is the embedding matrix that maps each word w_i into a dense vector x_i , W_f and f_i are the parameters and the hidden state for the forward LSTM, W_b, b_i are the parameters and the hidden state for the backward LSTM. We use zero vectors for the initial f_0 and b_{I+1} . The sentence embedding h is obtained by concatenating f_I and b_1 .

CNN with channel-based attention

Spatial attention is an important component of visual reasoning and language understanding systems that solve tasks such as image captioning [Xu et al., 2015, Lu et al., 2017, Chen et al., 2017a] and visual question answering [Zhu et al., 2016, Xu and Saenko, 2016, Ben-Younes et al., 2017]. We use a novel channel-based spatial attention mechanism that focuses on the important features whilst preserving the spatial relations.

To contrast our channel-based attention mechanism with the more common location-based attention we first describe the latter and its shortcomings. The location-based attention mechanism picks an averaged feature vector from a feature map, i.e. it uses a weighted sum of all cells of the feature map as the perceptual representation. Although the attention weights contain the information about where to look, the parts of the network after the attention mechanism lose the location information since the operation reduces the spatial dimensions to obtain the feature representation. In addition to losing the location information, the location-based attention mechanism lacks the ability to focus on particular filters in the feature vector which act as a pattern detectors (e.g. a color in low-level layers or an object feature in high-level layers). The location-based attention bases the action on the similarity between the language representation and a single averaged feature vector (i.e. outputs of all pattern detectors in the last layer).

We designed our CNN architecture (Figure 2.4) so that its channel-based attention

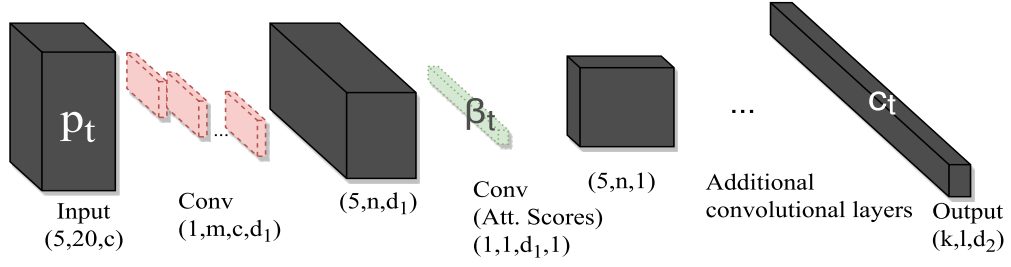


Figure 2.4: The convolutional neural network architecture with channel attention. The perceptual input p_t is a 5×20 grid and c is the length of the binary vector representation (Section 2.2.1) for each grid position. The first layer applies a $(1, m, c, d_1)$ filter bank, where $1 \times m$ is the size of the filter and d_1 is the number of filters. The output of the first layer is a $(5, n, d_1)$ tensor where $n = 20 - m + 1$. β_t is the attention vector determined by the decoder hidden state. After further convolutional layers, the output of the whole architecture, c_t , has (k, l, d_2) dimensions.

mechanism can focus on outputs of specific filters depending on the instruction and later layers can use the spatial information about where the focused filters have high activation. The first layer of the network applies a filter bank to the perceptual input p_t at time t (described in Figure 2.3) to detect objects/properties on it.

$$\text{out}_1 = \text{CONV}(W_{c1}, p_t) \quad (2.2)$$

where, W_{c1} is the filter bank and the CONV represents a convolutional layer¹.

Following the first layer, a decoder-controlled attention vector weighs the properties captured on the output channels of this first layer (out_1). The purpose of the attention vector is to direct perception to parts of the input relevant to the instruction. The combination of the first layer and the channel attention can be seen as picking a filter and applying it over the world. The resulting filter map allows higher layers to use the spatial information for further reasoning.

$$\text{out}_2 = \beta_t * \text{out}_1 \quad (2.3)$$

where $*$ is the convolution operation. We formulate the attention vector (β_t) as follows:

$$\beta_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^I \exp(e_{tj})} \quad (2.4)$$

¹Convolution, bias, and relu activation operations.

where β_{ti} is the weight of input channel i at time t . β_{ti} is calculated as the softmax of an attention score e_{ti} which is a function of the hidden state of the decoder s_{t-1} (defined in the decoder section):

$$e_{ti} = \text{NN}(W_a, s_{t-1}) \quad (2.5)$$

We use a feedforward neural network NN with parameters W_a and s_{t-1} as the input for the computation of the attention score. s_{t-1} conveys information from the instruction to direct the perception.

After the attention layer, we have further convolutional layers² to learn higher-order relations such as making comparisons among different directions or comparing/weighing the outputs for a specific direction.

$$\begin{aligned} \text{out}_3 &= \text{CONV}(W_{c2}, \text{out}_2) \\ c_t &= \text{CONV}(W_{c3}, \text{out}_3) \end{aligned} \quad (2.6)$$

where W_{c2} and W_{c3} are the corresponding filter banks, and c_t is the final output of the CNN architecture. We apply a relu activation after each convolution layer in the CNN except the last one, where we use the sigmoid function, giving the final perceptual state c_t .

Decoder

The decoder generates a sequence of actions $\mathbf{a} = (a_1, a_2, \dots, a_T)$ given the perceptual state c_t and the sentence embedding h . We implemented the decoder as a LSTM with hidden state s_t , where the initial hidden state is initialized by the sentence embedding: $s_0 = h$. It defines a probability distribution over the action sequence $\mathbf{a}$:

$$P(\mathbf{a}) = \prod_{t=1}^T P(a_t | s_t, c_t) \quad (2.7)$$

Our architecture models each conditional probability using the following formulation:

²Hyperparameter search led us to use one additional layer for SAILx dataset and two additional layers for SAIL dataset.

$$s_0 = h \quad (2.8)$$

$$s_t = \text{LSTM}(W_d, [c_t; a_{t-1}], s_{t-1}) \quad (2.9)$$

$$o_t = W_1 s_t + W_2 c_t + b \quad (2.10)$$

$$P(a_t | s_t, c_t) = \text{softmax}(o_t) \quad (2.11)$$

where p_t is the perceptual state at time step t , c_t is the output of the CNN with channel-based attention, h is the sentence embedding, s_t is the hidden state of the decoder LSTM with parameters W_d . The input of the LSTM is the concatenation of c_t and a_{t-1} . The unnormalized output o_t is a linear combination of c_t and s_t . The result of the softmax operation on o_t gives the conditional probabilities for possible actions.

2.2.3 Training

We use the cross-entropy loss to maximize the probability of the ground-truth action sequence. If the correct action sequence is $[a_1, a_2, \dots, a_T]$, we can train the model by minimizing the negative log-likelihood:

$$L = -\log P(a_1, a_2, \dots, a_T | \mathbf{w}, p_{1:T}) \quad (2.12)$$

$$= -\log \prod_{t=1}^T P(a_t | a_{1:(t-1)}, \mathbf{w}, p_{1:t}) \quad (2.13)$$

$$= -\log \prod_{t=1}^T P(a_t | s_t, c_t) \quad (2.14)$$

$$= -\sum_{t=1}^T \log P(a_t | s_t, c_t) \quad (2.15)$$

Note that the hidden state of the decoder, s_t , and the perceptual state c_t contain all the information from the actions $a_{1:(t-1)}$, the instruction $\mathbf{w}$, and the perceptual input $p_{1:t}$ used by the model. The weights of the model can be learned with backpropagation through time [Werbos, 1990].

2.2.4 Inference

Once we train the model, we generate action sequences by searching over alternative paths with beam search [Sutskever et al., 2014, Rush et al., 2015, Mei et al., 2016] using the distribution $P(a_t|s_t, c_t)$. For the *Paragraph* instances, we execute the beam search sentence by sentence, keeping the same beam between sentences. We generate actions for a sentence until the *STOP* action is taken, the agent hits a wall, or reaches a maximum number of actions. We also use an ensemble of trained models to obtain the action distribution by taking the average of the $P(a_t|s_t, c_t)$ distributions predicted by each model.

2.3 SAIL Dataset

We use the SAIL dataset published by [MacMahon et al., 2006]. In this dataset, there are 3 different maps (named *Grid*, *Jelly*, *L*). Each map consists of different number of nodes and edges. Nodes might have an item (barstool, chair, easel, hatrack, lamp or sofa). Halls have different floor patterns (blue, brick, concrete, flower, grass, gravel, wood or yellow) and different wall paintings (butterfly, fish or tower). Figure 2.1 depicts an example.

The dataset was generated by using two sets of participants, six instructors and thirty-six followers. The instructors studied the mazes until they were able to efficiently navigate. Then they were asked to give a list of instructions to navigate from a starting position to a goal position without looking at the map. All written errors in the instructions, both syntactic (typos, grammar errors, etc.) and semantic (confusing left and right, calling a chair as a sofa, etc.), remained as they were. The followers tried to follow the written list of instructions in the same maze without any prior knowledge of the environment. They were able to complete 69.64% of the paragraph-long instructions accurately.

[Chen and Mooney, 2011] split paragraphs into individual sentences and paired each sentence with the corresponding segment of the path. We call this version of the dataset *Single-Sentence* and the original one *Paragraph*. *Single-Sentence* contains 3237 instructions and *Paragraph* contains 706 sets of instructions each paired with their corresponding maps and paths.

To better understand the nature of the dataset, we analyzed the instructions in terms

Task and frequency	Definition	Examples
Language only (31.7%)	Instructions contain only linguistic dependencies. The perceptual information is not required to solve the task.	turn right, move two steps, turn right and take a step
Turn to X (7.0%)	The agent should understand the specified perceptual phenomenon and take one or two "turn" actions.	turn to the chair, turn to the green path
Move to X (13.4%)	The agent should understand the specified perceptual phenomenon and take one or more "move" actions.	move to the sofa, go to the end
Turn and Move to X (1.7%)	The agent should understand the specified perceptual phenomenon. The agent should first complete orientation step(s), then the movement(s).	turn and move to the chair, go to the easel
Orient (5.2%)	The agent should orient itself along the specified perceptual phenomenon. There might be more than one conditions.	turn so that the wall is on your back
Description (9.6%)	Instructions describe a specific agent-environment configuration. This type of instructions are generally used to describe the final position. The most of the instructions require to take only STOP action.	you should be at the intersection of blue and brown
Move Until (8.7%)	The agent should move in the forward direction until a specific perceptual phenomenon occurs.	walk forward until you reach the blue floors
Any combination (22.7%)	Instructions contain two parts and at least one of them has a perceptual dependency.	at the black road intersection take a left

Table 2.1: Instruction types in SAIL with their frequencies, definitions and examples.

of their perceptual and linguistic requirements. *Single-Sentence* instructions can be split into different categories: for example, language only instructions such as "turn left", where there is no need for perception, and ones such as "turn to the chair", which require perception. We give the definition, prevalence and examples for each task in Table 2.1. Although the dataset contains challenging language grounding problems, one-third of the data consists of language only instructions which do not require perceptual understanding. Even a model that does not use perceptual information is able to perform surprisingly well (Section 2.5.2).

Another drawback of the SAIL dataset is the lack of diversity for the action sequences that the agent should take to complete the instructions. Figure 2.5 shows that more than half of the instructions require a single movement followed by a stop action or only the stop action itself.

Given the problems with the dataset, machine learning models have a tendency to map specific instructions to specific action sequences due to the lack of diversity in the physical configurations (e.g. mapping "move to the chair" instruction to (MOVE, STOP) action sequence because the agent had always received that instruction when it was one step away

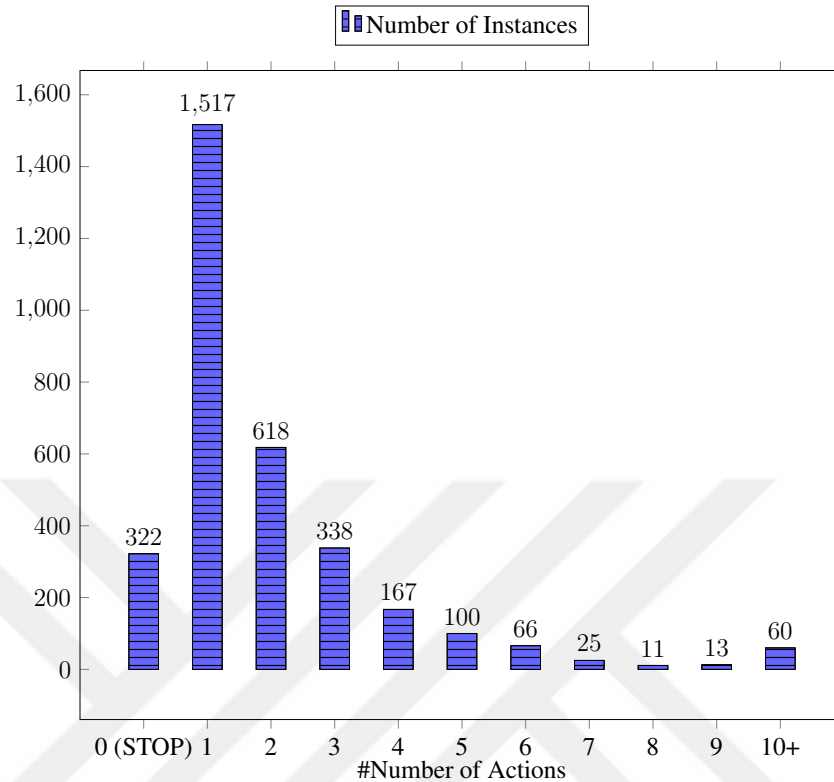


Figure 2.5: The distribution of the length of action sequences.

from the chair).

2.4 Synthetic Dataset (SAILx)

SAILx is a synthetic data generator that randomly generates maps, paths, and associated natural language instructions similar to the original SAIL data. Synthetic data generation allows us to investigate the problem of learning navigational instructions with fine-grained control over the map, the task, and the language. One can determine the complexity of the environment by controlling the different aspects such as the size of the map, length of the path, allowed items, flooring patterns, wall paintings and their locations. The language generation can be modified by controlling the vocabulary and text templates. The user chooses the category of each instance from the ones listed in Table 2.1. Thus one can generate datasets of different size and balance.

2.4.1 Map and Path Generation

We generate mazes (size of 8x8, in this study) using the recursive backtracker algorithm [Priestley and Ward, 1994] by selecting a random starting point. Once we obtain a maze, we decorate a random subset of nodes with random items. We divide the maze into two or three areas randomly and set the wall paintings of halls of each area with a distinct pattern. We choose the flooring patterns randomly but use the same pattern within a hall. We use the same set of objects, paintings and floor patterns as the SAIL dataset.

To generate a target path we select random start and goal points that are far enough (at least four steps in this study) from each other. We find the shortest path between the two points using the A^* algorithm [Hart et al., 1968]. If the generated path does not match with the queried task pattern, we reject the path.

2.4.2 Instruction Generation

We generate instructions in the *Single-Sentence* form using a large subset of the vocabulary from the original SAIL dataset. We segment a path into turning and moving parts and use one or two segments for each sentence. To generate the instructions, we use physical task patterns and pattern dependent text templates. As an example, for the *Move to X* task, "reaching an end" is a physical task pattern and "/move/go/walk to the end/wall" and "take the path/hall/corridor until the end/wall" are two text templates. We select the optional parts of a template randomly.

2.4.3 Instruction Coverage

We determined the task patterns and the text templates by examining the SAIL dataset and tried to generate instructions like the ones generated by human participants (See the Appendix A). Table 2.2 quantifies the proportion of the original instructions that our synthetic generator is able to generate (i.e. exact string matching). To measure this, we generate instructions for a specific task and compare them with the originals until the generator could not generate unique instructions in 5k consecutive attempts. Our generator can generate most of the instructions that occur more than once in the SAIL dataset.

Task	Frequency	Overall	Non-Unique
Language Only	31.70%	90.25%	98.11%
Turn to X	7.01%	41.41%	73.75%
Move to X	13.38%	33.72%	60.95%
Turn and Move to X	1.73%	50.00%	70.00%
Orient	5.16%	64.67%	94.50%
Description	9.64%	14.42%	70.83%
Move Until	8.71%	2.84%	42.86%
Any combination	22.67%	5.31%	23.86%

Table 2.2: The instruction types and coverage statistics for our generator. The second column shows the frequency of each task in the SAIL dataset. The third column gives the percentage of the original instructions SAILx is able to generate. The last column gives the percentage SAILx can generate for instructions that occur more than once.

2.4.4 The Fixed-Sized SAILx Dataset

For researchers who want to compare model performance on a fixed-sized dataset, we release a large set of pre-generated instructions with corresponding paths and maps. This dataset contains 160k instances compared to 3K in SAIL. Table 2.3 presents the number of instances for each instruction type in the fixed-sized SAILx dataset. The number of instances were chosen to allow our model to converge to 90% test set performance on all of the subtasks given in Table 2.1. The performance of our model on this dataset is given in Section 2.5.

2.5 Experiments & Results

We conducted experiments on both the SAIL and the SAILx datasets. First, we describe the baseline models used in the experiments. Then, we give the details of the experiments on the SAIL dataset and compare our model with the baselines as well as the state-of-the-art models. Finally, we explain our experiments on the SAILx dataset and our proposed evaluation methodology for model comparison.

Task	Number of Examples
Turn to X	20000
Move to X	20000
Turn and Move to X	15000
Language Only	25000
Move Until	15000
Any Combination	65000
Total	160000

Table 2.3: Task distributions in the fixed-sized SAILx data.

2.5.1 Baselines

Language Only (L.O.). This model is a regular sequence-to-sequence architecture without an attention mechanism. The encoder encodes the instruction with a bidirectional LSTM and the decoder predicts the action sequence taking the previous action as input at each time step. This model does not use any perceptual information. We use gold actions as the input of the decoder during training. At test time, we use the actions predicted by the decoder.

Bag-Of-Features (B.O.F). This baseline model is a simplified version of [Mei et al., 2016]. The encoder encodes the instruction with a bidirectional LSTM and the decoder predicts the action sequence taking the bag-of-features world state as input at each time step. This model does not contain any attention mechanism.

2.5.2 SAIL dataset

We experimented on both the *Single-Sentence* and the *Paragraph* version of the SAIL dataset. Following the previous studies, a trial is counted as successful if and only if the final position and orientation match with the ground-truth path for the *Single-Sentence* version. For the *Paragraph* version, matching the final position is sufficient for success.

We use the *Single-Sentence* version of the data for the training and we test the model on both *Single-Sentence* and *Paragraph* datasets. We use six fold cross validation for

Method	Single-Sentence	Paragraph
[Chen and Mooney, 2011]	54.40	16.18
[Chen, 2012]	57.28	19.18
[Kim and Mooney, 2012]	57.22	20.17
[Kim and Mooney, 2013]	62.81	26.57
[Artzi and Zettlemoyer, 2013]	65.28	31.93
[Artzi et al., 2014]	64.36	35.44
[Andreas and Klein, 2015]	59.60	-
[Kočíšký et al., 2016] †	63.25	-
[Mei et al., 2016] †‡	69.98	26.07
[Mei et al., 2016] ‡	71.05	30.34
[Fried et al., 2018] †‡	71.6	34.05
Human [MacMahon et al., 2006]	-	69.64
Our Model-vDev (avg)	67.3 (0.7)	24.1 (1.6)
Our Model-vDev (ens=10)	71.9	31.5
Our Model-vTest (avg)	69.9 (0.8)	27.5 (1.6)
Our Model-vTest (ens=10)	73.6	33.3
Our Model (avg)	68.7 (0.4)	25.6 (1.0)
Our Model (ens=10)	73.1	33.2

Table 2.4: Overall results. Results of the previous works were obtained using the “vTest” procedure if not explicitly stated otherwise. † denotes the vDev protocol. ‡ states that ensembling is used for the inference. The numbers in the parentheses are the standard deviations.

the tuning of the model using splits for each map (*Grid*, *Jelly*, *L*). We left two maps as training data and split the remaining one into development data (50%) and test data (50%). We repeat this experiment by swapping the test and development data. This process is carried out for each map as development/test data. We run each fold ten times with different seeds and report the size-weighted average of runs as the final score. We tune the hyper-parameters of the model depending on the average score of the six folds instead of tuning for each map.

The common approach on this dataset is using the test data for development, denoted by “vTest” below. This approach is problematic because of the usage of the test data in the tuning process. [Mei et al., 2016] proposed another approach (vDev) to tune the model by using 10% of the two maps as the development data and the remaining (90%) as the training data. This approach is also problematic because there may be linguistic differences among the instructors for different maps. By using half of the test map data for development and the other half for test, we avoid both problems.

We use Adam [Kingma and Ba, 2014b] with default parameters for the optimization and gradient clipping [Mikolov et al., 2010a] with the norm threshold 5. We use the success rate on the development data for early stopping. We stop the training if the model does not improve the development score for 10 consecutive iterations. We update the model after seeing each paired instruction and action sequence with no minibatching.

Table 2.4 shows the overall performance of our model against the previous studies with the “vTest” experimenting scheme. Our final model achieved the state-of-the-art result on *Single-Sentence* even though we do not tune the model for each map and use the test set for tuning. The performance on the *Paragraph* dataset is comparable with the best model [Artzi et al., 2014] which uses an additional semantic lexicon for the initialization of a semantic parser.

Table 2.5 presents results of the baseline models compared to our final model. The *Language Only* baseline achieves surprisingly good results on the visual tasks without receiving perceptual information. One reason for this is that some instructions for the visual tasks contain non-perceptual clues to solve the task, e.g. “turn **right** to the chair”, “move **forward two steps** to the easel”. The ability of the network to exploit the bias hidden

Task	L.O.	B.O.F.	Ours
Language Only	88.0	87.9	90.7
Turn to X	86.8	89.4	90.3
Move to X	64.9	69.6	80.1
Turn and Move to X	44.6	71.4	71.4
Orient	87.2	93.4	95.2
Description	93.4	89.4	88.1
Move Until	35.1	44.0	46.1
Any combination	19.9	32.8	37.9
Overall	64.2	69.5	73.1

Table 2.5: The performance (test accuracy) comparison of baselines and the proposed model. Results were obtained using ensemble of ten models for each architecture. Experiments have been done using the schema proposed in this study.

in the action sequence distribution (see Figure 2.5) might be another factor for the high performance of the language only model. For example, most of the action sequences are (*RIGHT*, *STOP*) in the *Orient* task.

The Bag-Of-Features baseline performs better than the Language Only baseline in almost all tasks. Since both models have the same architecture for instruction modeling, it is expected that they result in similar performance in the Language Only task which we observe. Our final model further improves the B.O.F baseline in most tasks.

2.5.3 SAILx dataset

Language understanding tasks like navigational instruction following contain instruction and task types of different complexity. Small fixed-sized datasets like SAIL may fail to differentiate the performance of models due to sparsity or imbalance of instruction types. Table 2.5, in particular the high performance of the language-only model, shows that the SAIL dataset fails to sufficiently differentiate the abilities of different models. At the other extreme, a very large fixed-sized dataset could fail to differentiate the sample efficiency of high capacity models of similar peak performance. We need better evaluation

methodologies to evaluate both the sample efficiency and peak performance of models.

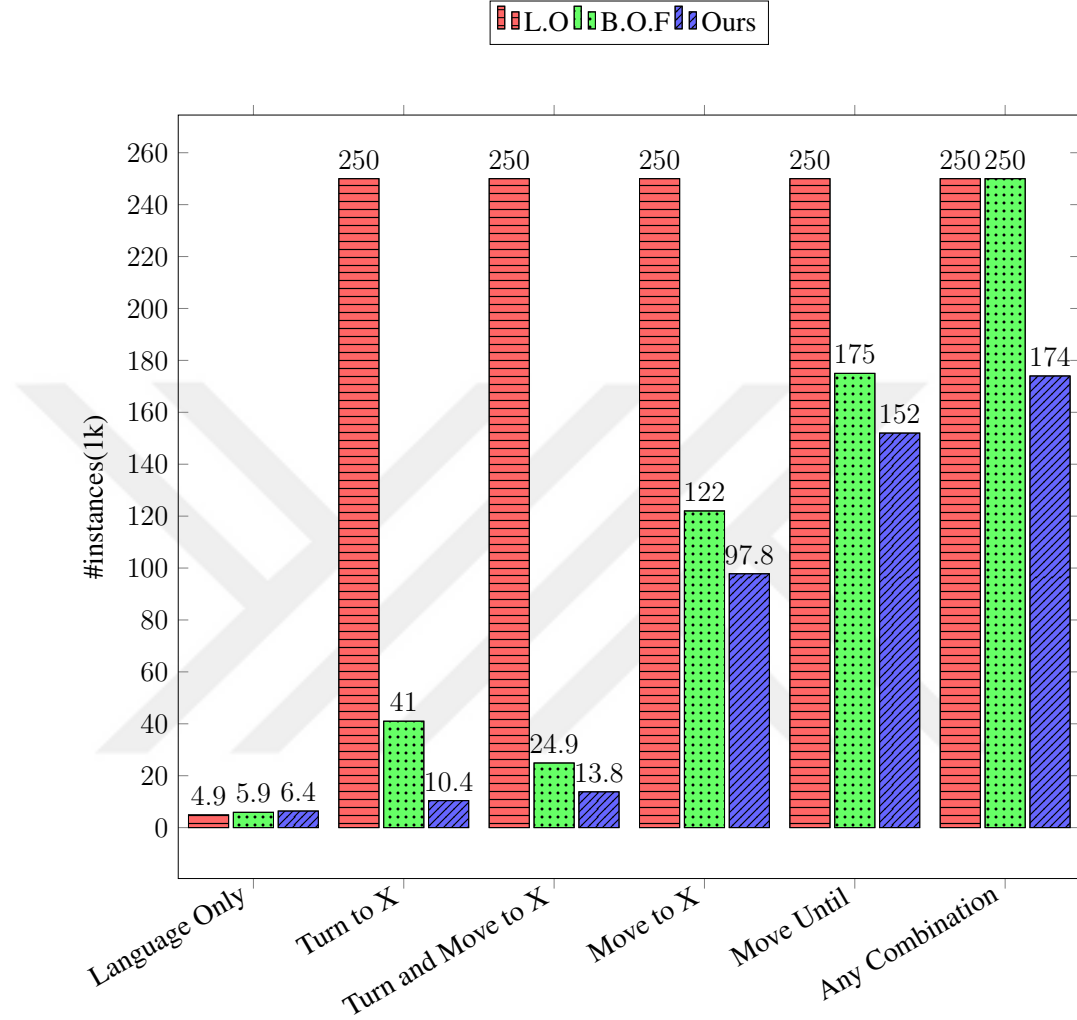


Figure 2.6: The learning efficiency comparison between the baseline models and the grid based model. The y axis shows the number of instances required to reach 90% accuracy on unseen data.

For this purpose, we used the SAILx synthetic data generator to compare the learning efficiency of the proposed model and the baseline models. We train each model on a stream of data for a specific task category until the test accuracy exceeds a threshold. The test accuracy is calculated as a moving average of accuracy on the next unseen instance³. We repeat this for each task from scratch to see whether a model fails for a specific task.

³moving_average = 0.95 * moving_average + 0.05 * average_accuracy_on_current_batch

Figure 2.6 illustrates the number of instances required to reach 90% test accuracy by the baseline models and our final model. We performed the hyper-parameter search for each model and for each task using golden section search [Kiefer, 1953]. We use Adam [Kingma and Ba, 2014b] with default parameters for the optimization. We use gradient clipping [Mikolov et al., 2010a] and set the norm of the gradients 5.

Our grid based model achieves better or equal efficiency in almost all tasks when it is compared to the performance of the B.O.F baseline. The B.O.F baseline is not able to solve the *Any Combination* task within the 250k instances limit. Additionally, both models show similar performance on the *Language Only* task because of their similar architecture for language processing. The L.O model can only solve the *Language Only* task and it is the most efficient model for this task.

In Table 2.6, we compare the performance of the baseline models and our model on the fixed-sized SAILx dataset (Section 2.4). We split this data into train (70%), dev (15%) and test (15%) sets while preserving the proportion of each subtask. Our agent-centric grid based model is significantly better than the B.O.F. baseline. The L.O. baseline reaches an accuracy higher than the proportion of language-only tasks. One reason might be that since the model uses the previous action as the input of the decoder, it captures some common action sequence patterns.

	Dev		Test	
	Avg. (std)	Ens.	Avg. (std)	Ens.
L.O	55.0 (0.2)	54.9	54.8 (0.2)	54.8
B.O.F	84.4 (0.2)	85.2	84.4 (0.1)	85.3
Ours	92.0 (0.6)	93.6	91.9 (0.6)	93.5

Table 2.6: Results on the generated fixed data. The average and ensemble scores were obtained by using 10 models for each architecture. The numbers in the parentheses are the standard deviations.

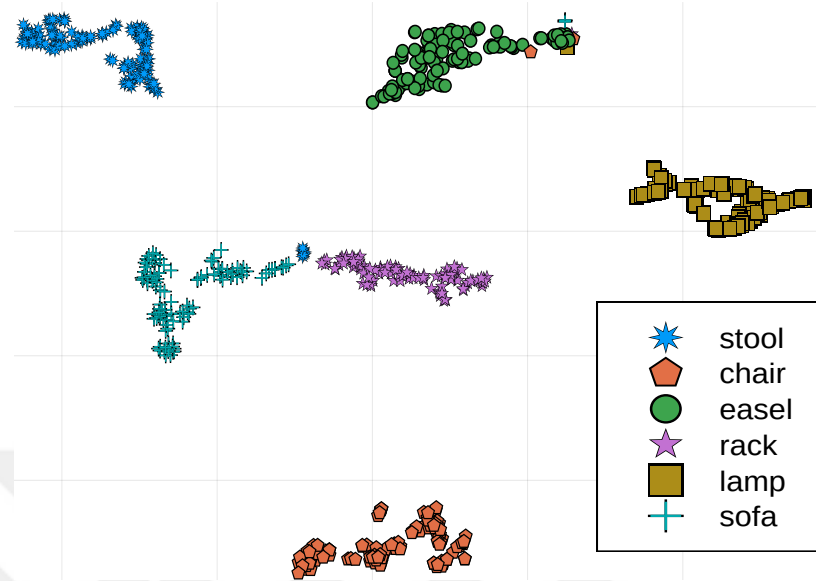


Figure 2.7: t-SNE visualization of attention weights.

2.5.4 Visualization of Attention Weights

The decoder in our model uses the language information provided in the sentence embedding from the encoder and uses this information to direct the perception. For this, the decoder queries the output of filters in the first layer of the CNN via the channel attention. Note that the visual input contains high level information (e.g object type, flooring pattern, etc.) about the environment, so the filters in the first layer would learn to detect those properties.

To see the relation between attention weights and the instructions, we clustered the attention weights using the t-SNE algorithm [Maaten and Hinton, 2008]. We obtained attention vectors for the first step of the instructions that belong to the *turn_to_x* category in the test split of the SAILx dataset. We also eliminated the instructions that do not refer to an object. Figure 2.7 illustrates that our model consistently focuses on outputs of similar filters for the similar instructions.

2.6 Related Work

We first summarize the studies on artificial data generation. Next, we outline the general literature on grounded language learning.

Artificial Data Generation

Similar to SAILx, SHAPEWORLD [Kuhnle and Copestake, 2017], SHAPES [Andreas et al., 2016b], CLEVR [Johnson et al., 2017a] datasets provide control over the language and the world to examine subproblems of the visual question answering problem. The bAbI tasks [Weston et al., 2015] also provide controlled subtasks, however, these are text only whereas our synthetic dataset is multi-modal.

[Yu et al., 2017] proposed an environment for navigating in a 2-D maze like environment, XWORLD. In contrast to the SAIL environment, in XWORLD, the environment is fully observable to the agent. [Chaplot et al., 2017] proposed a new 3D environment built on the ViZDoom API [Kempka et al., 2016]. They manually generated 70 navigation instructions where each instruction is a combination of (action, attribute(s), object) triple. [Hermann et al., 2017] proposed a simulated environment in 3-D. The language used in both environments involve only referring expressions to navigate the agent. [Chang et al., 2017] developed the MATTERPORT environment which is a simulation framework for photo-realistic indoor scenes. Recently, [Anderson et al., 2018b] proposed the Room-to-Room dataset which includes natural language instructions obtained from mechanical-turks for navigating an agent in the MATTERPORT environment. [Chen et al., 2019b] introduced TOUCHDOWN task and dataset which is based on routes and images from Google Street View.

In the aforementioned navigational instruction following tasks, the actions taken by the agent do not change the environment. This limits both linguistic and perceptual complexity of the tasks. Recently, [Misra et al., 2018] introduced the CHAI environment which combines both navigation and simple manipulation instructions. CHAI involves an “interact” action along with the navigation actions. Similarly, ALFRED [Shridhar et al., 2020] involves both manipulation and navigation instructions. ALFRED has more

manipulation actions than CHAI, which provides richer visual and linguistic challenges.

Grounded Language Learning

[MacMahon et al., 2006] published the SAIL dataset and demonstrated a rule based system that uses manually aligned language and world features. The most commonly studied approach in this domain is mapping natural language instructions into a formal meaning representation. [Chen and Mooney, 2011] presented a system that translates instructions to formal executable plans by training the KRISP semantic parser [Kate and Mooney, 2006] with aligned instruction and action-sequence pairs. Later, [Chen, 2012] improved the system by modifying the underlying semantic grammar and tested the system also on Chinese instructions. [Kim and Mooney, 2012] approached grounded language learning as probabilistic context free grammar (PCFG) induction introduced by [Börschinger et al., 2011] to the navigation domain. They further improved this technique using a re-ranking module with the task-specific weak signal [Kim and Mooney, 2013]. [Artzi and Zettlemoyer, 2013] learned a semantic parser seeded with a manual lexicon. The parser operates on a combinatory categorical grammar (CCG) to translate the instruction into a lambda-calculus formalism for the semantic representation. They improved their system using a re-ranker and controlling the size of the lexicon in a data driven fashion [Artzi et al., 2014].

Another approach is learning to translate the instruction into an action sequence in an end-to-end fashion similar to our model. [Andreas and Klein, 2015] modeled the instruction following task as scoring possible execution plans depending on the alignment with a given instruction. They used a conditional random field model to learn this alignment. [Mei et al., 2016] used a textual attention-based [Bahdanau et al., 2014] encoder decoder neural network in contrast to our model which has an attention mechanism over the channels for the visual attributes. Although [Mei et al., 2016] used attention to words, we observed that adding this mechanism does not bring any improvements. We suspect that initializing the decoder with the encoder’s last hidden state is enough to convey the linguistic information in the SAIL domain. Our agent-centric grid based representation for the perceptual states and channel based attention mechanism brings an improvement

over the bag-of-features representation of [Mei et al., 2016]. [Kočíský et al., 2016] utilized an auto-encoder objective to enable semi-supervised training by leveraging randomly generated unsupervised data (random action sequences). They used a similar model to the bag-of-features baseline and showed that the model can benefit from unsupervised training. [Fried et al., 2018] introduced a speaker-listener model where a pragmatic speaker aims to generate instructions that have less ambiguity and are more interpretable by the follower model. The pragmatic listener reasons about the speaker to select the action sequences that match with the speaker intention. They used the model of [Mei et al., 2016] as the base listener and showed that pragmatic inference provides significant improvements. Since their pragmatic inference does not depend on the base models, we believe that using our model as the base model may bring further improvements.

2.7 Conclusions

We developed a novel agent-centric grid based representation for the perceptual states of a navigation agent, in contrast with previous approaches where spatial relations are missing. We combined this representation with a novel channel based attention mechanism designed to more easily pair with language. Our model resulted in state-of-the-art accuracy in the SAIL *Single-Sentence* dataset without using any external resources. We developed a synthetic data generation framework in an artificial navigation domain that can be used to provide an unlimited number of perception-instruction-action instances to train and evaluate grounded language learning models. We proposed an evaluation methodology to measure the learning efficiency of a model using the number of instances to reach a particular performance in well specified tasks rather than the overall accuracy reached on a fixed-sized dataset.

Chapter 3

LEARNING FROM IMPLICIT INFORMATION IN NATURAL LANGUAGE INSTRUCTIONS FOR ROBOTIC MANIPULATIONS

3.1 Introduction

Consider yourself standing in your kitchen and having your robot assist you in preparing tonight's meal. You then give it the instruction: *'fetch the bowl next to the bread knife!'*. For the robot to correctly perform your intended instruction, which is grounded in your world representation, it must correctly ground your natural language instruction into its own world representation.

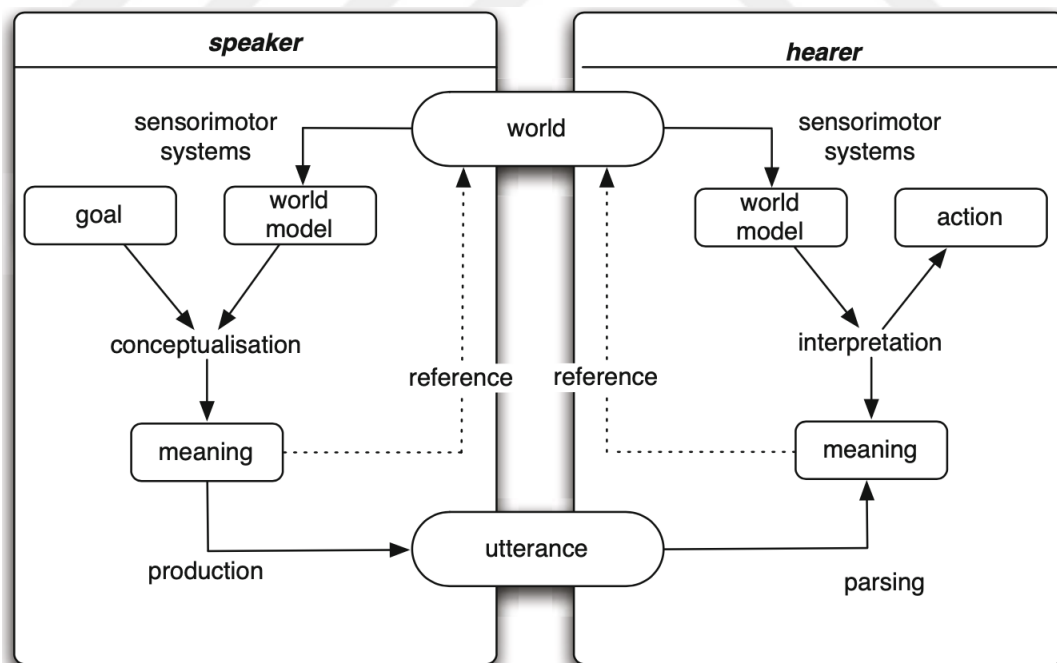


Figure 3.1: The semiotic cycle [Steels and Hild, 2012] shows the communication process between the speaker and the hearer.

[Steels and Hild, 2012] schematized the semiotic cycle as in Figure 3.1, in which speakers and hearers go through for the communication. In this chapter, we are interested in the hearer part which is a robot trying to understand and follow the instructions given by a user. The hearer must generate possible meaning candidates by parsing the utterance and then interpret these candidates by grounding onto its world representation. To be able to continue to communicate, the agent should maintain its world representation. This already introduces the two key components of language grounding in robotics: the construction of a world representation from sensor data and the grounding of natural language into the constructed representation. Ideally these two components would be learned in a joint fashion [Hu et al., 2017a, Johnson et al., 2017b, Santoro et al., 2017, Hudson and Manning, 2018, Perez et al., 2018]. However, the scarcity of data makes this approach impractical. The millions of data points necessary for state-of-the-art joint computer vision and natural language processing are simply non-existing. We opt, therefore, to separately learn the world representation component and the language grounding component.

Learning the World Representation. One approach for constructing a world representation of a robot is through so-called *perceptual anchoring*. Perceptual anchoring handles the problem of creating and maintaining, over time, the correspondence between symbols in a constructed world model and perceptual data that refer to the same physical object [Coradeschi and Saffiotti, 2000]. In this work, we use sensor driven bottom-up anchoring [Loutfi et al., 2005], whereby anchors (symbolic representations of objects) can be created by perceptual observations derived directly from the input sensory data. When modeling a scene, based on visual sensor data, through object anchoring, noise and uncertainties will inevitably be present. This leads, for example, to a green “apple” object being incorrectly anchored as a “pear”.

Language Grounding. We use the Neural Module Networks [Andreas et al., 2016b] for language grounding. This approach is suitable to the process described in Figure 3.1. A parser generates a computational graph from the instruction, and execution of this graph on the world representation yields the grounding. The computational graph consists of the learnable neural nodes which carry out the grounding functionality. We designed our own modules that suit the internal world representation. To learn these components, we use the

synthetic data that simulates the world represented as anchors. This means that we do not ground the language using sensor data as signal but a symbolic representation of the world. During training these symbols are synthetic and simulated, and during the deployment of the language grounding these are anchors provided by the anchoring system.

Belief Revision. As a result of learning the construction of a world model and the language grounding separately, **contradictions** arise **between** the world **representations** of a human and a robot. The supervision that an instruction would give to a robot is not present when learning the representation of the world of a robot. These inconsistencies then propagate through to inconsistencies between the instructions a human gives to a robot and the robot's world model. To ensure that a robot is able to correctly carry out an instruction, such inconsistencies must be resolved and the world model of the robot be matched to the world model of the human. [Gärdenfors, 2014] describes this communicators problem as "*meeting of minds*".

This is not the first study that tackles the problem of belief revision [Alchourrón et al., 1985, Gärdenfors, 1988, Gärdenfors, 1990] in robotics, where an agent updates its belief about the current state of the world with the respect to the new information about the objects or properties of the environment or the occurrences of events. However, prior work [Tellex et al., 2013, Thomason et al., 2015, She and Chai, 2017], with the notable exception of [Mast et al., 2016], relied on explicit information transfer between humans and robots when inconsistencies arose in grounded language and the robot's world representation. An example would be a robot asking clarification questions until it is clear what the human meant [Tellex et al., 2013].

One available clue, which does not require the external signal, to update the belief is hidden information in the instructions. Since the instruction already states the objects (e.g., bowl, mug) and their spatial relations (e.g., to the right, behind), the hearer can update the belief about the world towards the configuration that is consistent with the grounding by taking the speaker's world model as base. This idea is in line with the recent studies specify that language could improve the human visual system. [Lupyan and Ward, 2013] experimented whether comprehending a verbal label affects the visual detection of objects and showed that the label absence decreases the performance. Similarly, [Boutonnet et al.,

2013] showed that hearing a label improves the visual awareness of the salient objects. In addition, [Forder and Lupyan, 2019] demonstrated that hearing a color label increases the accuracy of distinguishing the named color from the colors nearby. To facilitate this idea, we propose an approach that probabilistically reasons over the grounding of an instruction and a robot’s world representation in order to perform Bayesian learning to update the world representation given the grounding. This is closely related to the work of [Mast et al., 2016] who also deploy a Bayesian learning approach. The key difference, however, is that they do not learn the language component but ground a description of a scene by relying on a predefined model to ground language.

We demonstrate the validity of our approach for reconciling instructions and world representations on a showcase scenario involving a camera, a robot arm and a natural language interface. We also show the generalization and compositionality of the designed modules with a simulated *Naming Game* [Steels, 1995]. Iteratively increasing the world complexity, we train the components to learn individual nouns, adjectives, prepositions and their compositions. We show that after training individual components, composing them requires almost no learning.

3.2 Preliminaries

The overarching objective of our system is to plan and execute robot manipulation actions based on natural language instructions. This requires, in the first place, that both the planner of the robot manipulator, as well as the natural language grounder (cf. Section 3.2.2), share a joint semantically rich object-centered model of the perceived environment, i.e., a *semantic world model* [Elfring et al., 2013].

3.2.1 Visual Object Anchoring

In order to model a semantic object-centered representation of the external environment, we rely upon the notions and definitions found within the concept of perceptual anchoring [Coradeschi and Saffiotti, 2000]. Following the approach for sensor-driven bottom-up acquisition of perceptual data, as described by [Persson et al., 2019], the used anchoring

procedure is, initially, triggered by sensory input data provided by a *Kinect2 RGB-D sensor*. Each frame of input *RGB-D* data is, subsequently, processed by a *perceptual system*, which exploits both the visual 2-*D* information, as well as the 3-*D* depth information, in order to: 1) detect and segment the subset of data (referred to as *percepts*), that originates from a single individual object in the physical world, and 2) measure *attribute values* for each segmented percept, e.g., measuring a *position attribute* as the $\mathbb{R}^3$ geometrical center of an object, or a visual *color attribute* measured as a color histogram (in *HSV* color space).

The percept-symbol correspondence is, thereafter, established by a *symbolic system*, which handles the grounding of measured attributes values to corresponding predicate symbols through the use of *predicate grounding relations*, e.g., a certain peak in a color histogram, measured as a *color attribute*, is mapped to a corresponding predicate symbol ‘red’. In addition, we promote the use of an *object classification* procedure in order to semantically categorize and label each perceived object. The convolutional neural network (CNN) architecture that we use for this purpose is based on the *GoogLeNet* model [Szegedy et al., 2015], which we have trained and fine-tuned based on 101 object categories that can be expected to be found in a kitchen domain.

The extracted perceptual and symbolic information for each perceived object is then encapsulated in an internal data structure α_t^x , called an *anchor*, indexed by time t and identified by a unique identifier x (e.g. ‘mug-2’, ‘apple-4’, etc.). The goal of an *anchoring system* is to manage these anchors based on the result of a *matching function* that compares the attribute values of an unknown candidate object against the attribute values of all previously maintained anchors. Anchors are then either created or maintained through two general functionalities:

- *Acquire* – initiates a new anchor whenever a candidate object is received that does not match any existing anchor α^x .
- *Re-acquire* – extends the definition of a matching anchor α^x from time $t - k$ to time t . This functionality assures that the percepts pointed to by the anchor are the most recent perceptual (and consequently also symbolic) representation of the object.

However, comparing attribute values of anchored objects and percepts by some distance

measure and deciding, based on the measure, whether an unknown object has previously been perceived or not is a non-trivial task. Nevertheless, since anchors are created or maintained through either one of the two principal functionalities *acquire* and *re-acquire*, it is evident that the desired outcome for the combined compared values is a *binary* output, i.e. should a percept be acquired or re-acquired. In previous work on anchoring [Persson et al., 2019], we have therefore suggested that the problem of invoking a correct anchoring functionality is a problem that can be approximated through learning from examples and the use of *classification algorithms*. For this work, we follow the same approach.

3.2.2 Natural Language Grounding

In this study, we focus on understanding spatial language that includes *pick up* and *place* related verbs, and *referring expressions*. An instruction refers to a target object using its representative features (e.g. color, shape, size). If a noun phrase does not resolve the ambiguity in the world, the instruction resolves the ambiguity by specifying the target object with its relative position to other surrounding objects. This hierarchy tries to bring the attention to finding the unique object, then shifts the attention to the targeted object. Based on this idea, we model the language grounding process as controlling the attention on the world representation by adapting the neural module networks approach proposed by [Andreas et al., 2016b].

Our natural language grounder has three components: a preprocessor, an instruction parser and a program executor. Given specific anchor information (Figure 3.2 – № 1), the preprocessor transforms the anchor information into an intermediate representation in grid form (Figure 3.2 – № 2). The instruction parser produces a computational program by exploiting the syntactic representation (Figure 3.2 – № 3) of the instruction with a dependency parser¹. The program executor runs (Figure 3.2 – № 4) the program on the intermediate representation to produce commands.

Preprocessor. The anchoring framework maintains the object descriptions predicted from the raw visual input. To be able to ground the language onto those descriptions, we map the available information (object class, color, size and shape attributes) to a 4D grid

¹<https://spacy.io/>

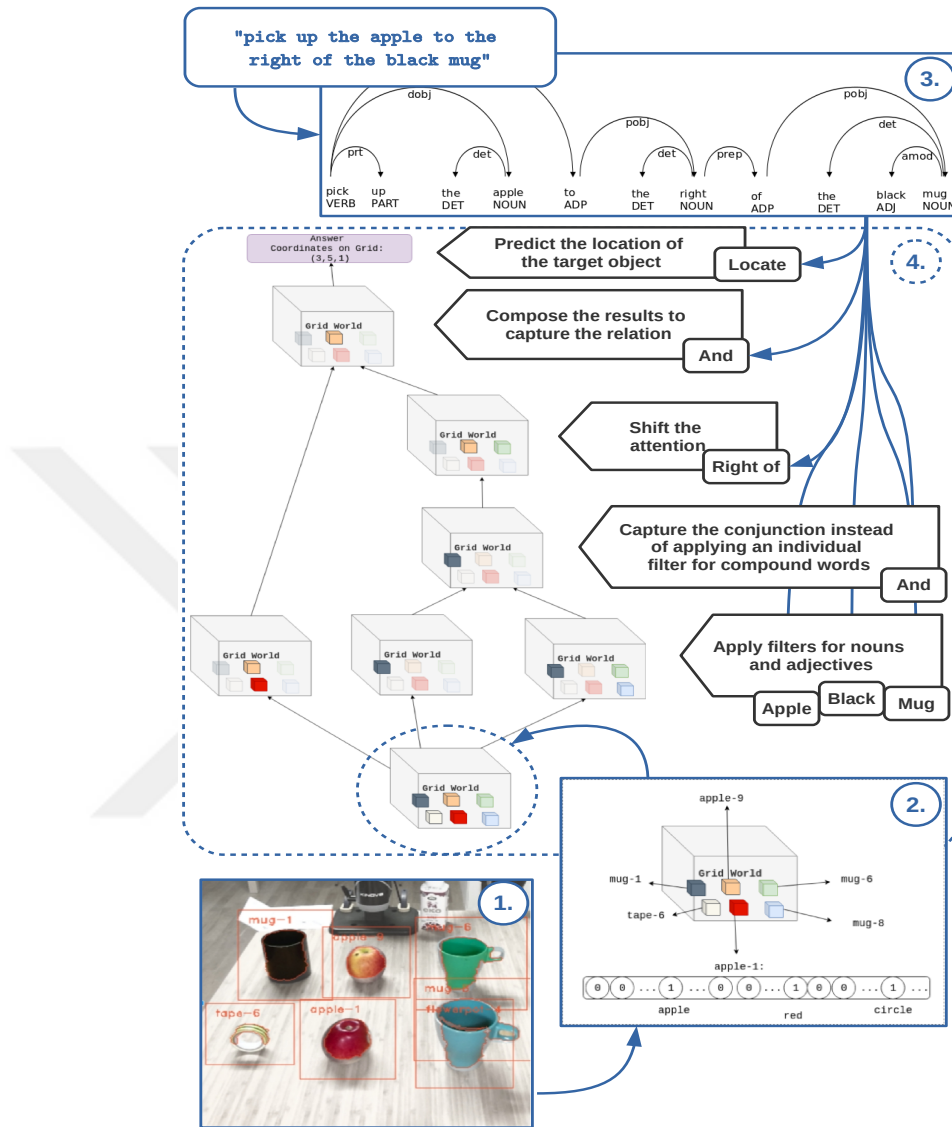


Figure 3.2: Demonstration of the language grounding process for the instruction “pick up the apple to the right of the black mug”. Anchoring system sends the snapshot of the anchors (1). Then, a preprocessor transforms the anchors into a grid representation which the language grounding system operates on (2). The parser parses the given instruction and generates a computation graph which specifies the execution order of neural modules (3). Finally, the neural modules are executed according to the computation graph to produce the action (4).

representation. We represent each anchor as a multi-hot vector and assign this vector to a cell where the real world coordinates of the object fall into.

Program Executor. The program generated by the parser is a collection of neural

components that are linked to each other depending on the computation graph. The design of neural components reflects our intuition about the attention control. A **Detect** module is a convolutional neural network with a learnable filter that captures a noun or an adjective. This module creates an attention map over the input grid.

$$Detect(w, b, x) = relu(w \otimes x + b) \quad (3.1)$$

The **Detect** module operates on the original grid input x tensor, where the dimensions are (W, H, L, C) . The first three dimensions represent the spatial dimensions and C denotes the length of the feature vector. w is the filter of size $(1, 1, 1, C)$ and b is the bias. $\otimes$ is a convolution operation.

Although a **Detect** module can capture the meaning of a noun phrase (e.g., red book), the model cannot generalize to unseen compound words. To overcome this, we design the **And** module to compose the output of incoming modules. This module multiplies the inputs element-wise in order to calculate the composition of words (e.g., the big red book). Since the incoming inputs are attention maps over the grid world, an **And** module produces a new attention map by taking the conjunction of its inputs. In the following equation, the $\odot$ denotes the element-wise multiplication.

$$And(a_1, a_2) = a_1 \odot a_2 \quad (3.2)$$

An output of a subgraph for a noun phrase is an attention map that highlights the positions for the corresponding objects that occur. A **Shift** module shifts this attention in the direction of the preposition that the module represents. This module is also a convolutional neural network similar to a **Detect** module. However, the module remaps the attention instead of capturing the patterns in the grid world.

$$Shift(w, a) = relu(w \otimes a) \quad (3.3)$$

The **Shift** module operates on an incoming attention map, where the dimensions are $(W, H, L, 1)$. w is the filter of size of $(2 * W + 1, 2 * H + 1, 2 * L + 1, 1)$. We use the padding to be able to perform the shifting operation over the whole grid. The pad size is the same as the input size.

A **Locate** module takes an attention map and produces a probability distribution over cells by applying a softmax classifier for being the targeted object. We use the cell with the highest probability as the prediction. A **Position** module gets a source anchor, a preposition and a target anchor, and produces a real world coordinate. It merely calculates the position available in the direction of the preposition from the target anchor, where the source anchor can fit.

Parser. We find the verbs in the instruction along with the subtrees attached to them. For each verb and its subtree, we search for the direct object of the verb. Then we build a subgraph for the direct object and its modifiers. Depending on the verb type, we build different subgraphs. If the verb is “pick up” related, then we look for the preposition that relates the given noun to another noun. If one is found, then a subgraph is created for the preposition object using the noun phrase that the object belongs to. Finally, the end point of the subgraph is combined with a *Shift* module. For each preposition object, we repeat the same process to handle prepositional phrase chains.

If the verb is “put” related, we find the preposition that is linked to the verb and the object of the preposition. We build a subgraph that refers to the object of the preposition similar to the “pick up” case. Finally, there is a *Position* module to produce the coordinates to put the direct object, where the position is referred with the auxiliary objects.

3.3 System Description

In the upper part of Figure 3.3, we illustrate our physical *kitchen table* system setup, which consists of the following devices: 1) a *Kinova Jaco light-weight manipulator* [Campeau-Lecours et al., 2019], 2) a *Microsoft Kinect2 RGB-D* sensors, and 3) a dedicated PC with an Intel® Core™ i7-6700 processor and an Nvidia GeForce GTX 970 graphics card.

In addition, we have a modularized software architecture that utilizes the libraries and communication protocols available in the Robot Operating System (ROS)². Hence, each of the modules, illustrated in the lower part of Figure 3.3, consists of one or several individual subsystems (or ROS nodes). For example, the *visual object anchoring* module

²<http://www.ros.org/>

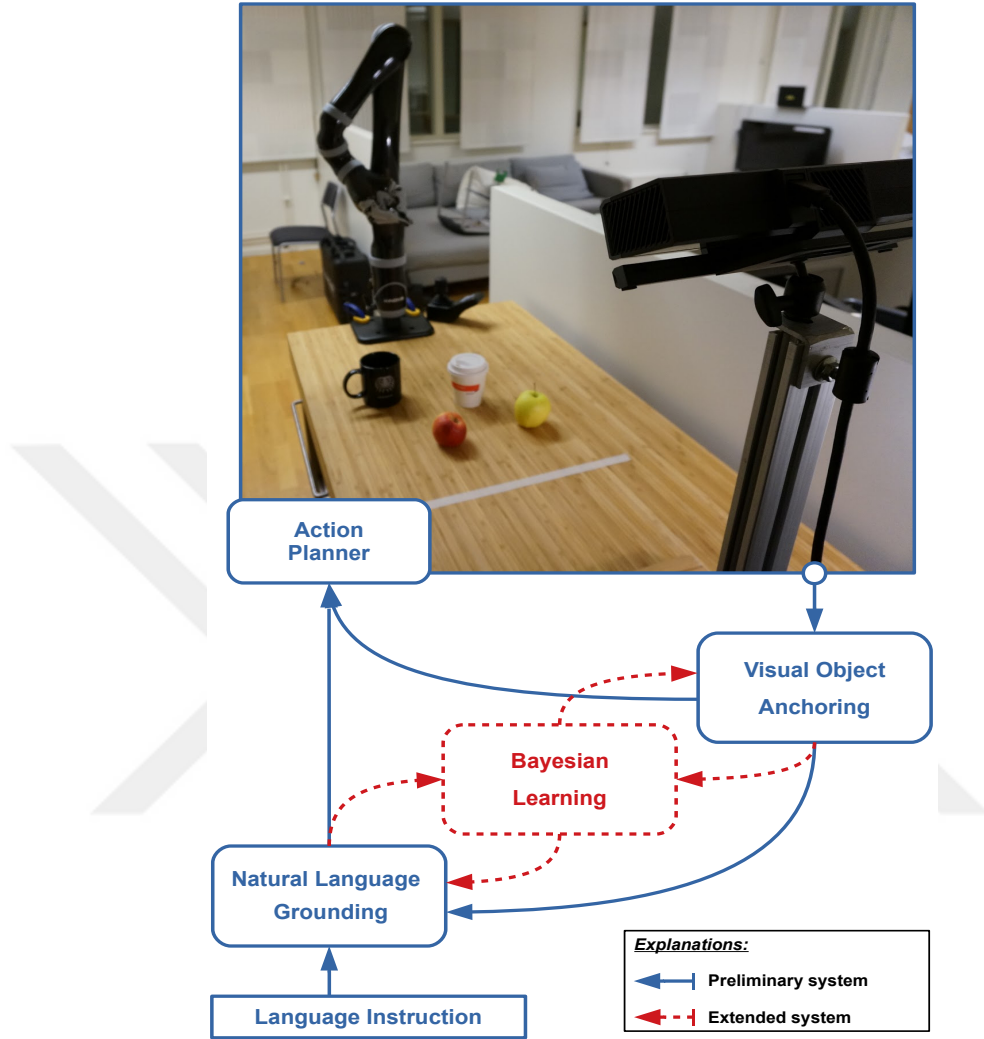


Figure 3.3: A depiction of both used physical system setup (upper), as well as used software architecture (lower). The arrows represent the flow of data between the modules of the software architecture. Blue solid arrows and boxes illustrate the preliminary system (outlined in Section 3.2), while red dashed arrows and boxes illustrate the novel extension for reasoning about different symbolic label configurations (and hence resolving inconsistencies between language and perception), by using Bayesian learning (as presented in Section 3.4).

consists of the following subsystems: 1) a *perceptual system*, 2) a *symbolic system*, and 3) an *anchoring system*. For a seamless integration between software and hardware, we are further taking advantage of both the *MoveIt! Motion Planning Framework*³, as well as the

³<https://moveit.ros.org/>

ROS-Kinect2 bridge developed by [Wiedemeyer, 2015]. The *MoveIt!* "planning scene" of the *action planner* for the robot manipulator, as well as the grid world representation used by the language grounding system (cf. Section 3.2.2), are, subsequently, both populated by the same updating anchoring representations (cf. Section 3.2.1). Hence, the visual sensory input stream is indirectly mapped to both objects considered in the dialogue by the language grounder, as well as the objects upon which actions are executed.

3.4 Resolving Inconsistencies

Based purely on the perceptual input, the anchoring system produces a probability distribution $p(l)$ over the possible labels (e.g. $[0.65 : \text{apple}, 0.35 : \text{pear}]$) for each anchor. We are now interested in the probability of a label l for an anchor given a natural language instruction i and the grounding g of that instruction in the real world. This is the conditional probability $p(l \mid g, i)$. We introduce, furthermore, the notion of a label configuration c . This is easiest explained by an example: imagine having two anchors and each of the anchors has two possible labels, then there are 2×2 possible label configurations. A label configuration is, hence, a label assignment to all the anchors present in the scene.

Now we need to transform the conditional probability into a function that is computable by the anchoring system and the language grounder. The first steps (Equations 3.4-3.6) are quite straight forward and follow basic probability calculus.

$$p(l|g, i) = \sum_c p(l, c|g, i) \quad (3.4)$$

$$= \sum_c p(l|c, g, i) p(c|g, i) \quad (3.5)$$

$$= \sum_c p(l|c) p(c|g, i) \quad (3.6)$$

In Equation 3.6 we assume that g and i are conditionally independent of the label of an anchor given the label configuration c . This can be seen in the following way. Imagine two anchors with two possible labels each. Given that we are in a specific label configuration, we immediately know what label the single anchors have. This means that the probability of a label for an anchor is 1 if it matches the label in the configuration and 0 otherwise. This reasoning is independent of the grounding and the instruction.

We have now split up the labels (produced by the anchoring system) and the grounding into two factors, which can be calculated separately. The first one can be calculated as follows:

$$p(l \mid c) = \frac{p(l, c)}{p(c)} = \frac{\prod_{j \in c} p(l_j)}{N_c} \quad (3.7)$$

This is the product of the probabilities of the labels that constitute a label configuration divided by the number of configurations. Assuming a uniform distribution over the label configurations (division by N_c) is equivalent to assuming that each possible label configuration is equally likely *a priori*. This means that we make no assumption about which class of objects occur more regularly or which class of objects (of the 101 possible classes) occur more often together with other classes of objects.

We tackle now the second factor in Equation 3.6. Equation 3.8-3.11 are again straightforward probability calculus. In Equation 3.12 we assume that the label configuration and the instruction are independent: their probabilities factorize. In Equation 3.13 the probabilities of i cancel out and we assume again a uniform distribution for the label configurations (cf Equation 3.7). In Equation 3.14 we then have a numerator and denominator that are expressed in terms of $p(g \mid c, i)$, which is exactly the function approximated by our neural language grounding system, cf. subsection 3.2.2.

$$p(c|g, i) = \frac{p(c, g, i)}{p(g, i)} \quad (3.8)$$

$$= \frac{p(g|c, i)p(c, i)}{p(g, i)} \quad (3.9)$$

$$= \frac{p(g|c, i)p(c, i)}{\sum_c p(c, g, i)} \quad (3.10)$$

$$= \frac{p(g|c, i)p(c, i)}{\sum_c p(g|c, i)p(c, i)} \quad (3.11)$$

$$= \frac{p(g|c, i)p(c)p(i)}{\sum_c p(g|c, i)p(c)p(i)} \quad (3.12)$$

$$= \frac{p(g|c, i)^{1/N_c}}{\sum_c p(g|c, i)^{1/N_c}} \quad (3.13)$$

$$= \frac{p(g|c, i)}{\sum_c p(g|c, i)} \quad (3.14)$$

Plugging Equations 3.7 and 3.14 back into Equation 3.6 gives the learned probability

of the label l of an anchor given the instruction i and the grounding g of that instruction.

$$p(l \mid g, i) = \sum_c \frac{(\prod_{j \in c} p(l_j)) p(g \mid c, i)}{N_c \sum_c p(g \mid c, i)} \quad (3.15)$$

As mentioned in Section 3.2.1 the anchoring system encapsulates 101 object categories, which means that the anchoring system produces a categorical probability distribution over 101 different labels for each anchor. With only two anchors this results in already 101^2 different configurations. It is easy to see that computing $p(l \mid g, i)$ (cf. Equation 3.15) suffers from this curse of dimensionality. Therefore, we limited ourselves to the two labels with the highest probability per anchor, in the experiments too. This gives 2^{N_A} possible configurations, with N_A being the number of anchors present.

3.5 Experiments

3.5.1 Synthetic Data

Data demanding nature of neural networks requires large amounts of data to generalize well. Artificial data generation is one way of generating such datasets [Andreas et al., 2016b, Kuhnle and Copestake, 2017, Johnson et al., 2016]. Therefore, we designed a series of artificial learning tasks as “*Naming Game*” [Steels, 1995] before applying the model to a real-world problem. In each task, we generate a random grid world that provides the necessary complexity and ambiguity that fit the scenario. First, an object is placed on the grid world and decorated with attributes randomly as the target object. Then depending on the scenario, an auxiliary object and distractors (objects that have similar attributes as the target object) are placed on the grid world. We also generate objects that are not related to the target (or auxiliary object) to introduce additional noise. We limit the total number of objects to 10. We set the number of distractors as 2 in the experiments. Finally, we generate the ground truth computation graph for composing neural modules. We list the scenarios below in increasing order of difficulty (i.e., a combination of the ambiguity present in the grid world and the number of language components involved), which is in line with the order of the word acquisition in young children [Parish-Morris et al., 2013]. In a cross-lingual study, [Bornstein et al., 2004] demonstrate that nouns and adjectives occur before than the prepositions in the vocabulary of young children. Similarly, the

below curriculum aims to teach initially nouns, adjectives and then the prepositions. Next, the curriculum provides examples to test the compositionality of the learned abilities.

1. Using the **name** of a targeted object in the instruction is enough to localize the targeted object.
2. There is more than one object that has the same category with a targeted object. To solve the ambiguity, one or more discriminative **adjective(s)** are used.
3. The same world configuration as the second one. To solve the ambiguity, the object is described with a **prepositional phrase** that utilizes a single referent object.
4. The same world configuration as the third. **Adjectives** are used to describe a targeted object in addition to a **prepositional phrase**. In this case, adjectives are unnecessary, but the scenario measures whether additional components bring noise or not.
5. All other objects that have the same category with a targeted object have the same set of features as the targeted object has. Hence, the targeted object is only distinguishable by its position. To solve the ambiguity, the object is described with a **prepositional phrase** that utilizes a referent object along with necessary **adjectives**.
6. It is a **random** scenario from the above list.

3.5.2 Training

To be able to measure the compositionality of learned modules, we have two different settings for the data generation. For training, we constrain the 75% of possible attributes for an object class and locations on the grid world that an instance of that object class can present. During testing, we use unconstrained samples generated for the same scenario. This way, we can evaluate if the model infers unseen word compositions, e.g. inferring red mug after seeing red book and black mug in the training time. We follow a curriculum schema to train our modules. Starting from the first scenario described in Section 3.5.1, we train the model on a stream of constrained randomly generated samples. We evaluate

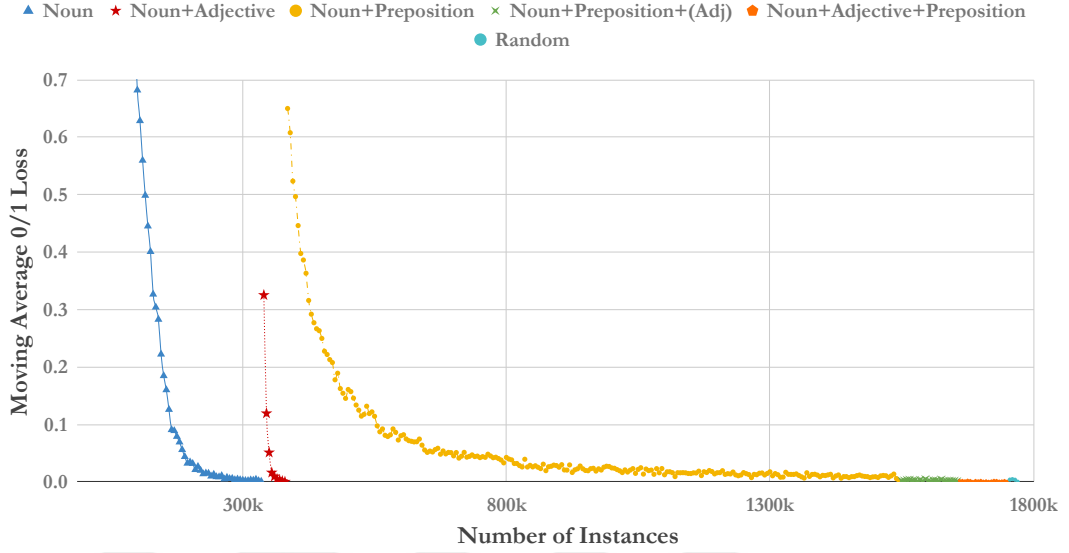


Figure 3.4: Learning curve of the neural modules.

the model periodically on unconstrained samples generated for each period and continue training until the moving average error on the test data falls under a threshold (e.g. $1e-5$ in our experiments). We then continue to train the model for the next scenario using learned weights. We set the number of nouns, adjectives and prepositions as 102, 26, and 27, respectively to match with the anchoring system. We use Adam [Kingma and Ba, 2014a] with default parameters (i.e. $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) for the optimization.

Figure 3.4 presents the learning curve of the model. The third graph (yellow) demonstrates that learning prepositions requires more data as compared to learning nouns (first graph) or adjectives (second graph). The reason for this behavior is twofold. First, the *Shift* modules have more weights to be learned than the *Detect* modules. Second, while the *Detect* modules have a one to one mapping between input and output, the *Shift* modules have many to many relations. There might be more than one active area in the input of a *Shift* module. Since it needs to remap highlighted areas on the grid to other areas, it needs to see different examples that occur in different parts of the grid world in order to learn to ignore the position of the active area.

The remaining graphs show the effectiveness of our design to compose learned modules.

Since we do not train any new modules from scratch for the corresponding experiments, the graphs show that the model require almost no new data, which indicates that the model is able to use the trained modules as they are without adjusting them for the composition of nouns, adjectives and prepositions.

3.6 Showcase

We now proceed with a demonstration of the integrated system: we have a Kinect camera that observes the world, the anchoring representation that builds up a representation of the world based on the raw image data, the language grounder that takes as input a natural language instruction and a probabilistic reasoning component that resolves possible inconsistencies between the robot’s world representation and the instruction.

The physical setup up is identical to the one depicted in the image in Figure 3.3: the robot arm is mounted on the opposite site of a kitchen table of the Kinect camera. The natural language instruction is passed to the language grounder via an *instruction prompt*. In each of the four panels in Figure 3.5, the instruction prompt is seen at the bottom as rectangular box. We further describe the scenario in the caption of Figure 3.5.

3.7 Related Work

Our work is related to two research domains: modular neural nets for language grounding and human-robot interaction for handling ambiguities in one or more modalities. [Andreas et al., 2016b, Andreas et al., 2016a] introduced neural module networks for visual question answering. [Johnson et al., 2017b, Hu et al., 2017a] developed policy gradient based approaches to learn to generate layouts instead of using a dependency parser based method. [Hu et al., 2017b, Yu et al., 2018a, Cirik et al., 2018] applied modular neural networks approach on “Referring Expression Understanding” task. [Das et al., 2018] demonstrated the usage of neural module networks in decision taking in a simulated environment. To our knowledge, the present study is the first work that uses neural module networks approach in the real-world robotic setting.

Learning from human interaction has been extensively studied. [Lemaignan et al.,

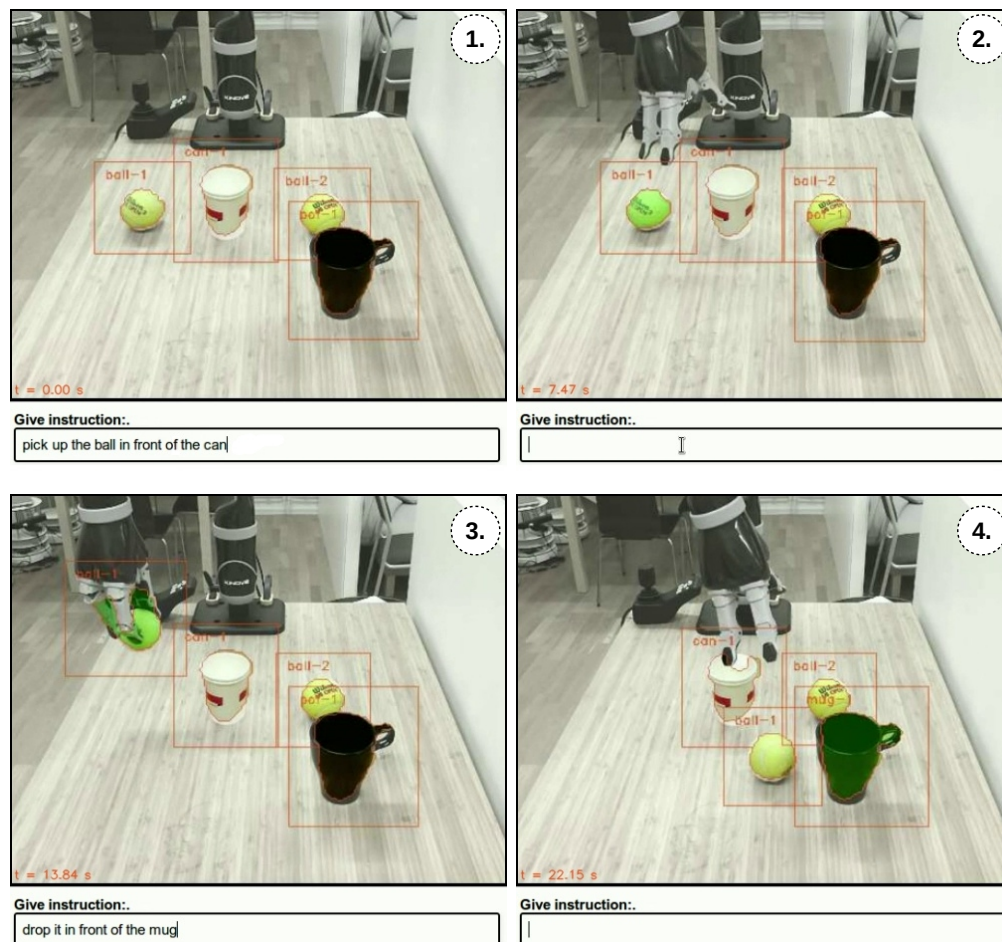


Figure 3.5: We give the robot the instruction: “*pick up the ball in front of the can*”. The robot executes the action and waits for further instructions. We then give the instruction to “*drop it in front of the mug*”. The problem in this step is that there is no object classified as ‘mug’, which means that none of the objects has as label with highest probability ‘mug’. We correct for this through probabilistic reasoning over not only the top label for each object but a number of top ranked labels per object. This allows the anchoring system to correct its classification of an object based on what we as humans think an object is. Given the instruction, the anchoring system re-classifies the black object from ‘pot’ to ‘mug’. The instruction is then successfully carried out. The recorded video can be found here: <https://vimeo.com/302072685>.

2011, Lemaignan et al., 2012] developed a cognitive architecture that makes decisions by using symbolic information provided as facts (pre-defined) or extended via human-robot dialogues. When compared with our system, their system neither operates on the sensory input nor deals with the uncertainty in the world. [Tellex et al., 2013] proposed a system

to ask questions to disambiguate the ambiguities presented in the instructions. The robot decides the most ambiguous part of the command which is defined based on a metric derived from entropy and asks questions about it to reduce the uncertainty. They update the generalized grounding graph [Kollar et al., 2013] with answers obtained from the user and use these to perform inference. In contrast, we fix the ambiguity present in the perceptual data. [She and Chai, 2017] proposed a system to learn to ask questions during the learning of verb semantics. They work on the *Tell me Dave environment* [Misra et al., 2014]. The work represents the environment as grounded state fluents (i.e. a weighted logic representation). In this work, language grounding is modeled as the difference between before and after state for an action sequence. They modeled the interactive learning as an MDP and solved it with reinforcement learning.

[Thomason et al., 2015] proposed a system that learns the meaning of natural language commands through human-robot dialog. They represent the meaning of instructions with λ -calculus semantic representation. Their semantic parser starts with an initial knowledge and learns through training examples generated by the human-robot conversations. Their dialog manager is a static policy which generates questions from a discrete set of action, patient, recipient tuples. [Padmakumar et al., 2017] improved this work with a learnable dialog manager. They train both the dialog manager and the semantic parser with reinforcement learning. This approach was further extended in [Thomason et al., 2019], where the authors combine the approach in [Thomason et al., 2015] and [Thomason et al., 2017] to obtain a system that is capable of concept acquisition through clarification dialogues. Instead of asking questions, we implicitly fix the perception with the information hidden in instructions. A further difference to these works is that we learn the language component in a simulated offline step, whereas they deploy active online learning, starting from a limited initial vocabulary.

This is also related to the work of [Perera and Allen, 2015], who present a system that tries to emulate child language learning strategies by describing scenes to a robot agent, which has to learn actively new concepts. The authors deploy probabilistic reasoning to manage erroneous sensor readings in the vision system. Apart from the active learning approach, there is also a conceptual difference: in our work, we do not consider discrepan-

cies between the perceptual system (anchoring) and the language grounder as errors in the perceptual system but simply as different models of the world.⁴

As mentioned in Section 3.1, the work related closest to our approach is presented in [Mast et al., 2016]. The authors base their work on geometric *conceptual spaces* [Gärdenfors, 2004], which situates their work in the sub-domain of *top-down anchoring* [Coradeschi and Saffiotti, 2000]. The geometric conceptual spaces induce a probabilistic model-based language grounder. This enables a robot to reason probabilistically over a description of a scene, given by an other agent, and single out the object that is most likely being referred to. In contrast, we present an approach to perform Bayesian learning over a learned language grounding model and a bottom-up anchoring approach.

3.8 Conclusions

In this study, we presented a robotic system that learns the world representation and the language grounding separately for natural language instruction following. We used the Neural Module Networks approach in a robotic system for the first time by allowing the system to learn the modules on a simulated data. We showed the generalization power and the learning efficiency of the approach on a Naming Game. We introduced the problem of belief revision in robotics based solely on implicit information available in natural language in the setting of sensor-driven bottom-up anchoring in combination with a learned language grounding model. This is in contrast to prior works, which study either explicit information or are based on top-down anchoring. We proposed a Bayesian learning approach to solve the problem and demonstrated its validity on a real world showcase involving computer vision, natural language grounding and robotic manipulation.

A main limitation of our current approach is the limited size of the predefined vocabulary. It would be more practicable if a robot were able to extend its vocabulary through the interaction with a human, i.e. through dialogue. A possible solution would be to learn a probabilistic model (which resolves inconsistencies between language and vision)

⁴This view taps into the philosophical question of whether one can ever truly know the nature of an object, cf. *thing-in-itself* [Kant, 1878], for which we omit a discussion.

that takes into account the possible of currently unknown vocabulary occurring. Such an approach would allow us to learn from the incomplete or ambiguous instructions by making use of the dialogue history.



Chapter 4

LANGUAGE CONTROLS MORE THAN TOP-DOWN ATTENTION: MODULATING BOTTOM-UP VISUAL PROCESSING WITH REFERRING EXPRESSIONS

4.1 Introduction

In this chapter we investigate whether language should be used to control the filters for bottom-up visual processing as well as top-down attention. In the human visual system, attention is driven by both “*top-down*” cognitive processes (*e.g.* looking for a specific object) and “*bottom-up*” salient, behaviourally relevant stimuli (*e.g.* fast moving objects) [Corbetta and Shulman, 2002, Connor et al., 2004, Theeuwes, 2010]. Studies on embodied language explore the link between linguistic and perceptual representations [Pulvermüller, 1999, Vigliocco et al., 2004, Gallese and Lakoff, 2005] and it is often assumed that language has a *high-level* effect on perception and drives the “*top-down*” visual attention [Bloom, 2002, Jackendoff, 2002, Dessalegn and Landau, 2008]. However, recent studies from cognitive science point out that language comprehension also affects low-level visual processing [Meteyard et al., 2007, Boutonnet and Lupyan, 2015]. Motivated by this, we propose a model that can modulate either or both of “*bottom-up*” and “*top-down*” visual processing with language conditional filters.

Current deep learning systems for language-vision tasks typically start with low-level image processing that is not conditioned on language, then connect the language representation with high level visual features to control the visual focus. To integrate both modalities, concatenation [Malinowski et al., 2015], element-wise multiplication [Malinowski et al., 2015, Lu et al., 2016, Kim et al., 2016] or attention from language to vision [Xu et al., 2015, Xu and Saenko, 2016, Yang et al., 2016, Lu et al., 2017, Anderson et al., 2018a, Zellers et al., 2019] have been used. Specifically they do not condition

low-level visual features on language. One exception is [De Vries et al., 2017] which proposes conditioning the ResNet [He et al., 2016] image processing network with language conditioned batch normalization parameters at every stage. Our model differs from these architectures by having explicit “*bottom-up*” and “*top-down*” branches and allowing us to experiment with modulating one or both branches with language generated kernels [Li et al., 2017, Gavriluk et al., 2018, Misra et al., 2018].

We evaluate our proposed model on the task of *image segmentation from referring expressions* where given an image and a natural language description, the model returns a segmentation mask that marks the object(s) described. We can contrast this with purely image based object detection [Girshick, 2015, Ren et al., 2017] and semantic segmentation [Long et al., 2015, Ronneberger et al., 2015, Chen et al., 2017b] tasks which are limited to predefined semantic classes. Our task gives users more flexibility to interact with the system by allowing them to describe objects of interest in free form language. The language input may contain various visual attributes (e.g., color, shape), spatial information (e.g., “on the right”, “in front of”), actions (e.g., “running”, “sitting”) and interactions/relations between different objects (e.g., “arm of the chair that the cat is sitting in”). This makes the task both more challenging and suitable for comparing different strategies of language control.

The perceptual module of our model is based on the U-Net image segmentation architecture [Ronneberger et al., 2015]. This architecture has clearly separated bottom-up and top-down branches which allows us to easily vary what parts are conditioned on language. The bottom-up branch starts from low level visual features and applies a sequence of contracting filters that result in successively higher level feature maps with lower spatial resolution. Following this is a top-down branch which takes the final low resolution feature map and applies a sequence of expanding filters that eventually result in a segmentation mask at the original image resolution. Information flows between branches through skip connections between contracting and expanding filters at the same level. We experiment with conditioning one or both of these branches with language.

To make visual processing conditional on language, we add language-conditional filters at each level of the architecture, similar to [Misra et al., 2018]. Our baseline only

applies language-conditional filters on the top-down branch. Modulating only the top-down/expanding branch with language means the high level features extracted by the bottom-up/contracting branch cannot be language-conditional. Our model expands on this baseline by modulating both branches with language-conditional filters. Empirically, we find that adding language modulation to the bottom-up/contracting branch has a significant positive improvement on the baseline model. Our proposed model achieves state-of-the-art performance on four different English referring expression datasets.

4.2 Model

Figure 4.1 shows an overview of our proposed architecture. For a given referring expression S and an input image I , the task is predicting a segmentation mask M that covers the object(s) referred to. First, the model extracts a $64 \times 64 \times 1024$ tensor of low-level features using a backbone convolutional neural network and it encodes the referring expression S to a vector representation r using a long short-term memory (LSTM) network [Hochreiter and Schmidhuber, 1997b]. Starting with the visual feature tensor, the model generates feature maps in a contracting and an expanding path where the final map represents the segmentation mask, similar to U-Net [Ronneberger et al., 2015]. 3×3 convolutional filters generated from the language representation r (language kernels) are used to modulate both the contracting and the expanding paths. Our experiments show that modulating both paths improves the performance dramatically.

4.2.1 Low-level Image Features

Given an input image I , we extract visual features from the fourth layer of the DeepLab ResNet101-v2 network [Chen et al., 2017b] pre-trained on the Pascal VOC dataset [Everingham et al., 2010]. We set $W = H = 512$ as the image size for our experiments. Thus, the output of the fourth convolutional layer of DeepLab ResNet101-v2 produces a feature map with the size of $(64, 64)$ and 1024 channels for this setup. We concatenate 8-D location features to this feature map following previous work [Hu et al., 2016b, Liu et al., 2017, Ye et al., 2019, Chen et al., 2019a]. The first three dimensions of the location

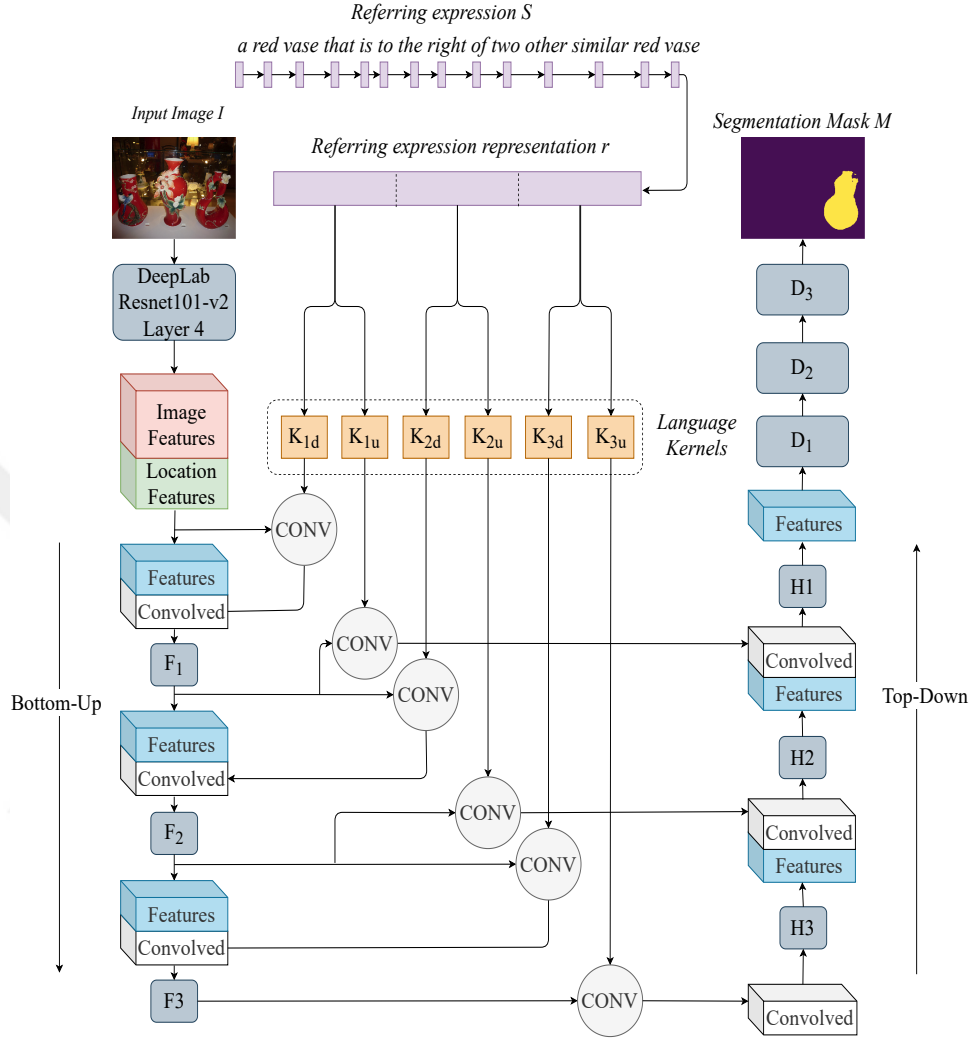


Figure 4.1: Overview of our model.

features represent normalized horizontal positions, the following three dimensions stand for vertical positions, and the last two dimensions represent the normalized width and height information of the image. The final representation, I_0 , has 1032 channels, and the spatial dimensions are $(64, 64)$.

4.2.2 Language Representation

Consider a referring expression $S = [w_1, w_2, \dots, w_n]$ where w_i represents the i 'th word. In this work, each word w_i is represented with a 300-dimensional GloVe embedding [Pennington et al., 2014], i.e. $w_i \in \mathbb{R}^{300}$. We map the referring expression S to hidden

states using a long short-term memory network [Hochreiter and Schmidhuber, 1997b] as $h_i = LSTM(h_{i-1}, w_i)$. We use the final hidden state of the LSTM as the textual representation, $r = h_n$. We set the size of hidden states to 256, i.e. $h_i \in \mathbb{R}^{256}$.

4.2.3 Segmentation Model

After generating image (I_0) and language (r) representations, our model generates a segmentation Mask M . We take the U-Net [Ronneberger et al., 2015] image segmentation model as the visual processing backbone. Our model extends the U-Net by conditioning both contracting and expanding branches on language using spatial language kernels.

Our model applies m convolutional modules to the image representation I_0 . Each module, F_i , takes the concatenation of the previously generated feature map ($Down_{i-1}$) and its convolved version with a 3×3 language kernel K_{id} and produces an output feature map ($Down_i$). Each F_i has a 2D convolution layer followed by batch normalization [Ioffe and Szegedy, 2015] and ReLU activation function [Maas et al., 2013]. The convolution layers have 5×5 filters with $stride = 2$ and $padding = 2$ halving the spatial resolution, and they all have the same number of output channels.

Following [Misra et al., 2018], we split the textual representation r to m equal parts (t_i) to generate language conditional convolutional filters (language kernels). Previous works preferred to use 1×1 dynamic language filters to integrate language and vision [Li et al., 2017, Misra et al., 2018, Gao et al., 2018]. Although 1×1 kernels are sufficient to pick out locations with a certain property (e.g. expressed by nouns and adjectives), in preliminary experiments we found that 3×3 kernels work better at capturing the relations between multiple locations (e.g. expressed by prepositions like “on”, “under”, “next to”). Therefore, we use each t_i to generate a spatial 3×3 kernel (K_{id}):

$$K_{id} = \text{AFFINE}_i(\text{DROPOUT}(t_i)) \quad (4.1)$$

Each AFFINE_i is an affine transformation followed by normalizing and reshaping to convert the output to a convolutional filter. DROPOUT is the dropout regularization [Srivastava et al., 2014]. After obtaining the kernel, we convolve it over the feature map

obtained from the previous module to relate expressions to image features:

$$G_{id} = \text{CONVOLVE}(K_{id}, \text{Down}_{i-1}) \quad (4.2)$$

Then, the concatenation of the resulting text-modulated features (G_{id}) and the previously generated features (Down_{i-1}) is fed into module F_i for the next step.

In the expanding branch, we generate m feature maps starting from the final output of the contracting branch as follows:

$$G_{ju} = \text{CONVOLVE}(K_{ju}, I_j) \quad (4.3)$$

$$Up_m = H_m(G_{mu}) \quad (4.4)$$

$$Up_j = H_j(G_{mu} \oplus Up_{j-1}) \quad (4.5)$$

Similar to the bottom-up phase, G_{ju} is the modulated feature map with language kernels generated as follows:

$$K_{ju} = \text{AFFINE}_j(\text{DROPOUT}(t_j)) \quad (4.6)$$

where AFFINE_j is again an affine transformation followed by normalizing and reshaping. Here, we convolve the kernel (K_{ju}) over the feature maps from the contracting branch (Down_j). Each upsampling module H_m gets the concatenation ($\oplus$) of the text related features and the feature map (Up_j) generated from the previous module. Only the first module operates on just convolved features. Each H_j consists of a 2D deconvolution layer followed by a batch normalization and ReLU activation function. The deconvolution layers have 5×5 filters with $\text{stride} = 2$ and $\text{padding} = 2$ doubling the spatial resolution, and they all have the same number of output channels.

After generating the final feature map Up_1 , we apply a stack of layers ($D_1, D_2, \dots, D_m$) to map Up_1 to the exact image size. Similar to upsampling modules, each D_k is a 2D deconvolution layer followed by batch normalization and the ReLU activation. The deconvolutional layer has 5×5 filters with $\text{stride} = 2$ and $\text{padding} = 2$ to double the spatial sizes of the input. Each D_k preserves the number of channels except for the last one which maps the features to a single channel for the mask prediction. There is no batch norm operation and the ReLU activation for the final module, instead we apply a sigmoid function to turn the final features into probabilities ($P \in \mathbb{R}^{H \times W}$).

4.2.4 Learning

Given the probabilities ($P \in \mathbb{R}^{H \times W}$) for each pixel belonging to the target object(s), and the ground-truth mask $G \in \mathbb{R}^{H \times W}$, the main training objective is the pixel-wise Binary-Cross-Entropy (BCE) loss:

$$J = \frac{1}{HW} \sum_i^H \sum_j^W G_{ij} \log(P_{ij}) + (1 - G_{ij}) \log(1 - P_{ij}) \quad (4.7)$$

If we pad the input image to reach a fixed size, we ignore the padded pixels during the loss calculation.

Multi-Scale Loss: Similar to [Lee et al., 2015, Chen et al., 2016, Dou et al., 2017], we use auxiliary pixel-wise BCE loss calculations for our final objective. Instead of using the loss value obtained from only the final layer, we also calculate auxiliary loss values after each module’s output in the expanding branch of the network. To this end, similar to the final layer, we use a deconvolution layer that maps the feature maps to probabilities. For each auxiliary loss calculation, we scale the ground-truth mask to match the size of the output generated from the corresponding layer. Our final objective is the sum of each loss term scaled by the ratio with respect to the original image size (e.g. 1, 0.5, 0.25, ...).

4.3 Experiments

In this section we first give the details of the datasets and our experimental configurations (Section 4.3.1). A detailed analysis of the contribution of our idea and the different parts of the architecture is given in Section 4.3.2. Then we present our main results and compare our model with the state-of-the-art (Section 4.3.3). Finally, Section 4.3.4 shows some qualitative results.

4.3.1 Datasets and Experiment Setup

We evaluate our model on four standard referring expression segmentation datasets and using two evaluation metrics.

UNC: This dataset [Yu et al., 2016] contains 142,209 referring expressions with 19,994 images obtained from MS COCO dataset [Lin et al., 2014]. Examples are collected interactively using a two-player game [Kazemzadeh et al., 2014] for 50k objects. Images can contain two or more instances of the same object class. There is no restriction on the expressions.

UNC+: Similar to UNC dataset, UNC+ [Yu et al., 2016] is also collected from MS COCO dataset. There are 141,564 referring expressions for 49,856 objects. The total number of images is 49,856. The difference between UNC and UNC+ is that location-specific expressions are not allowed in UNC+. Specifically, annotators are instructed to refer to an object by describing its appearance.

Google-Ref (G-Ref): This dataset [Mao et al., 2016] is also constructed from 26,711 images of MS COCO. There are 104,560 referring expressions for 54,822 objects. Expressions are collected from Amazon Mechanical Turk instead of a two-player game. This resulted in longer (avg. 8.43 words) and richer expressions than ones in UNC and UNC+ (avg. 4 words). Images contain 2-4 objects of the same class.

ReferItGame (ReferIt): The ReferIt dataset [Kazemzadeh et al., 2014] contains 130,525 expressions for 96,654 objects in 19,894 images. The images are from the IAPR Tc-12 dataset [Escalante et al., 2010]. In this dataset, there are also some expressions referred to stuff class such as water, sky, and ground. Due to the collection process, the expressions are shorter (avg. 3.4 words) in this dataset [Liu et al., 2017].

Evaluation Metrics: Following the previous work [Liu et al., 2017, Margffoy-Tuay et al., 2018, Ye et al., 2019, Chen et al., 2019a], we use overall intersection-over-union (*IoU*) and *precision@X* as evaluation metrics. Given the predicted segmentation mask and the ground truth, the *IoU* metric is the ratio between the intersection and the union of the two. The overall *IoU* calculates the total intersection over total union score. The second metric,

Method	<i>prec@0.5</i>	<i>prec@0.6</i>	<i>prec@0.7</i>	<i>prec@0.8</i>	<i>prec@0.9</i>	<i>IoU</i>
BRINet [Hu et al., 2020]	71.83	65.05	55.64	39.36	11.21	61.35
LSCM [Hui et al., 2020]	70.84	63.82	53.67	38.69	12.06	61.54
Top-Down	66.4	58.59	49.35	36.01	13.42	58.06
Bottom-Up	71.4	65.14	57.36	45.11	19.04	60.74
Dual (w/ FiLM)	71.18	65.14	57.32	44.66	18.75	61.12
Dual	75.12	70.08	63.32	50.05	22.29	63.59
Dual (nFilters=128)	75.36	69.8	63.04	49.14	21.46	63.24
Dual (nFilters=256)	74.92	69.72	62.82	49.6	21.54	63.09
Dual (nFilters=512)	75.12	70.08	63.32	50.05	22.29	63.59

Table 4.1: Ablation results and comparison with the previous works on the val set of UNC dataset with *prec@X* and *IoU* metrics.

precision@X, calculates the percentage of test examples that have *IoU* score higher than the threshold X . In experiments, $X \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

Implementation Details: As [Liu et al., 2017, Margffoy-Tuay et al., 2018, Ye et al., 2019, Chen et al., 2019a], we limit the maximum length of referring expressions to 20. In all convolutional layers, we set the filter size, stride, and number of filters (*ch*) as (5, 5), 2, and 96, respectively. The depth is 4 in the U-Net part of the network. We set the dropout probability to 0.2 throughout the network. We use Adam [Kingma and Ba, 2014b] with default parameters for the optimization. We do not train the DeepLab ResNet101-v2 weights. There are 60 examples in each minibatch. We train and test the model separately for each dataset. We train our model for 15 epochs on a Tesla V100 GPU (32GB) and each epoch take at most two hours depending on the train set’s size. We present the scores for the model that achieves the best performance on the validation split.

4.3.2 Ablation Results

We performed ablation studies to better understand the contributions of the different parts of our model. The bottom of Table 4.1 shows the performances of the different architectures

on the UNC validation split with $prec@X$ and overall IoU metrics.

Modulating both top-down and bottom-up visual processing: We implemented three models, Top-down Modulation, Bottom-Up Modulation and Dual Modulation, to show the effect of modulating language in expanding and contracting visual branches. Since language information leaks through cross-connections between visual branches, we also experimented with a bottom-up modulation model which has no connection between visual branches. Bottom-up Modulation outperforms Top-down Modulation with ≈ 2.68 IoU improvement. Modulating language in both visual branches yields the best results by improving Bottom-up Modulation model with ≈ 2.85 IoU score.

FiLM layers vs. Language-conditional Filters: Another method for modulating language is using conditional batch normalization [De Vries et al., 2017] or its successor, FiLM layers. Thus, when we replace language-conditional filters with FiLM layers in Dual Model we observe ≈ 2.4 IoU decrease. This shows the effectiveness of language kernels.

Number of Language-conditional Filters: We observed slight gain when we use more filters. Since increasing the number of filters brings more expressive richness, we decided to set the number of filters to 512 in the remaining experiments.

Multi-Scale Loss: Using only the predictions from final layer for the loss calculation decrease the performance slightly (≈ 0.6 IoU). We also observed that using multi-scale loss provides faster convergence on some datasets. For example, the model gets the best performance after only the sixth epoch on G-Ref dataset.

Shuffling the Training Data: Since we heavily use the batch norm operation in our architecture, to effectively train our model, we shuffle the training data after each epoch ends, as suggested in the Batch Normalization study [Ioffe and Szegedy, 2015]. Otherwise, we observe a performance drop (≈ 2 IoU), which would indicate that the model tends to memorize the minibatch statistics.

Method	UNC			UNC+			G-Ref	ReferIt
	val	testA	testB	val	testA	testB	val	test
CNN+LSTM [Hu et al., 2016a]	-	-	-	-	-	-	28.14	48.03
RMI [Liu et al., 2017]	45.18	45.69	45.57	29.86	30.48	29.5	34.52	58.73
DMN [Margffoy-Tuay et al., 2018]	49.78	54.83	45.13	38.88	44.22	32.29	36.76	52.81
DynamicFilters [Li et al., 2017]	-	-	-	-	-	-	-	54.30
KWA [Shi et al., 2018]	-	-	-	-	-	-	36.92	59.09
RRN [Li et al., 2018]	55.33	57.26	53.93	39.75	42.15	36.11	36.45	63.63
CMSA [Ye et al., 2019]	58.32	60.61	55.09	43.76	47.6	37.89	39.98	63.80
CAC [Chen et al., 2019c]	58.90	61.77	53.81	-	-	-	44.32	-
Step-ConvRNN [Chen et al., 2019a]	60.04	63.46	57.97	48.19	52.33	40.41	46.4	64.13
BRINet [Hu et al., 2020]	61.35	63.37	59.57	48.57	52.87	42.13	48.04	63.46
LSCM [Hui et al., 2020]	61.47	64.99	59.55	49.34	53.12	43.50	48.05	66.57
MAttNet [Yu et al., 2018b]	56.51	62.37	51.70	46.67	52.39	40.08	n/a	-
NMTree [Liu et al., 2019a]	56.59	63.02	52.06	47.40	53.01	41.56	n/a	-
Our Model	63.59	66.44	60.33	51.27	56.27	43.01	50.16	64.66

Table 4.2: Comparison with the previous works on four datasets. Evaluation metric is the overall *IoU* and higher is better. Bold scores indicate the state-of-the-art performances. “-” indicates that the model has not been evaluated on the dataset. “n/a” indicates that splits are not same.

4.3.3 Quantitative Results

Table 4.2 shows the comparison of our model with the previous work. Our model performs significantly better on G-Ref, UNC, and UNC+ datasets. G-Ref can be seen as the most complicated dataset since it contains the longest expressions (8.4 vs 3.4/4) and having more objects of the same class. UNC+ is also a more challenging dataset comparing to UNC, because the more semantic descriptions are required to be used to distinguish the objects. This shows that our model efficiently grounds the expression onto images by modulating the visual branches with spatial language filters.

The previous state-of-the-art method, LSCM, uses the external dependency parsing trees to obtain expression level multi-modal features by softly aggregating the word level multi-modal features. In contrast to this, our model only uses the generated language filters from the final hidden state of the LSTM to modulate visual processing with language. Our

model’s better performance on the more challenging (in terms of language complexity) datasets shows that our model is able to make use of the spatial language filters to model relations without explicit language structure knowledge. We see the same pattern when we compare CMSA and CAC. Although CAC uses a single layer to fuse visual features with language filters, it outperforms CMSA on the G-Ref dataset where CMSA appends the embedding of each word to the visual feature maps. They show comparable performances on the UNC dataset.

We also compare our model with MAttNet and NMTree which are referring expression comprehension models. Since they present segmentation results after they predict bounding boxes for objects, they are comparable with our work. Our model is significantly better than MAttNet, NMTree which depends on an explicit object proposal network that is trained on more COCO images. This result shows the ability of our model to detect object regions and relate them with expressions.

Table 4.1 presents the comparison of our model with the state-of-the-art in terms of $prec@X$ scores. The difference between our model and the state-of-the-art increases when the threshold increases. This indicates that our model is better at both finding and segmenting the referred objects.

4.3.4 Qualitative Results

In this section, we visualize some of the segmentation predictions of our model to gain better insights about the trained model.

Figure 4.2 shows some of the cases that our model segments correctly. These examples demonstrate that our model can learn a variety of language and visual reasoning patterns. For example, the first two examples of the first row show that our model learns to relate superlative adjectives (e.g., *longer*, *shorter*) with visual comparison. Examples include spatial prepositions (e.g., *on right*, *on left*, *next to*, *behind*, *over*, *bottom*) demonstrate the spatial reasoning ability of the model. We also see that the model can learn a domain-specific nomenclature (*catcher*, *batter*) that is present in the dataset. Lastly, we can see that the model can distinguish the different actions (e.g., *standing*, *squatting*, *sitting*).

We also analyzed the behaviour of our model with respect to incrementally given

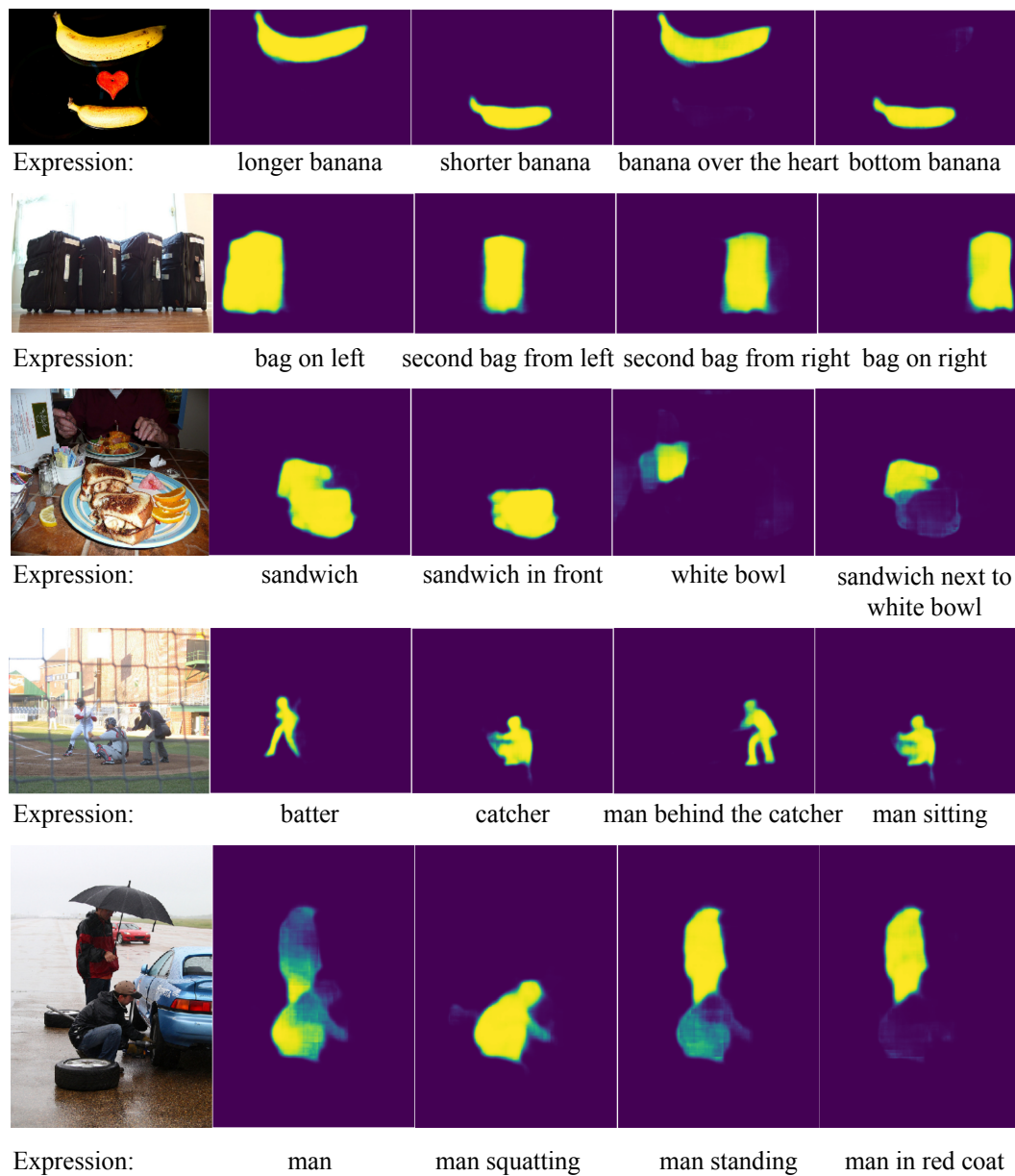


Figure 4.2: Visualization of some of the correct predictions of our model. First column shows the original images and others show the output of the model for the given referring expressions. Examples are taken from the UNC validation set.

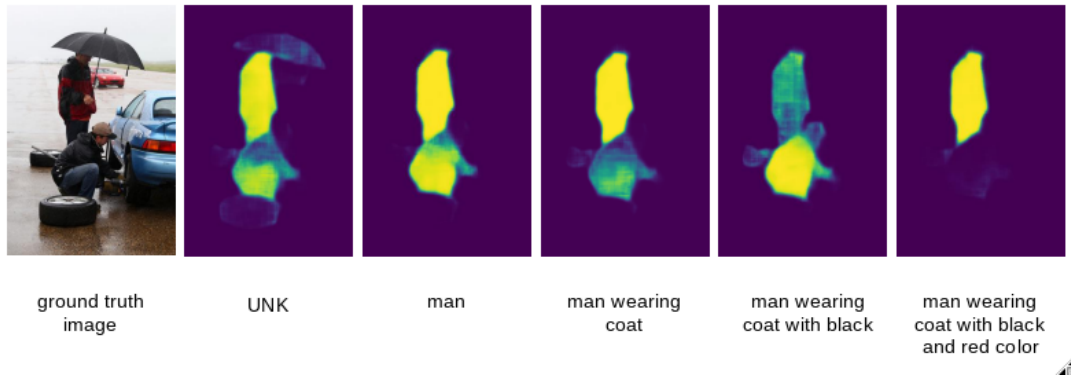


Figure 4.3: Incremental segmentation result of our model on UNC test split instance.

language input in Figure 4.3. In the initial step, our model only sees an unknown word token. In the second step, our model sees only the first word of the expression. In every step, our model starts to see a new word in addition to the previous ones. Figure 4.3 shows that our model can capture ambiguities in input image and expressions pairs. For unknown token input, our model captures all salient objects since there is no restriction. When the *man* word is fed, the model discards unrelated objects like umbrella and wheel. Additionally, when our model starts to see color words for coat, it initially focuses on both men, since both coats has black color. When it sees the final expression, it shifts its focus to the correct object.

Figure 4.4 shows some of the incorrect segmentation predictions from our model on the UNC validation dataset. In the figure, each group shows one of the observed patterns within the examples. One of them (a) is that our model tends to combine similar objects or their parts when they are hard to distinguish. Another reason for the errors is that some of the expressions are ambiguous (b), where there are multiple objects that could be the correspondence of the expression. And the model segments both possible objects. Some of the examples (d) are hard to segment completely due to the lack of light or objects that mask the referred objects. Finally, some of the annotations contain incorrect or incomplete ground-truth mask (c).

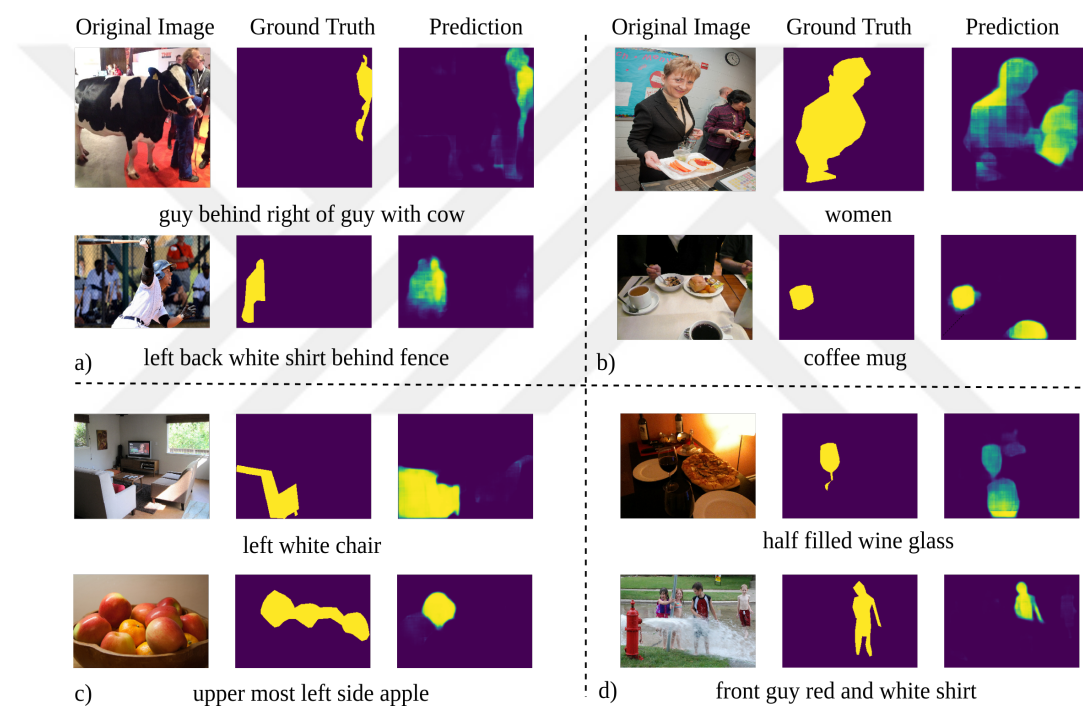


Figure 4.4: Visualization of some of the incorrect predictions of our model. Each group (a-d) shows one of the pattern we observed within the predictions. In each group, the first column shows the original image, the second one is the ground truth mask and the third one is the prediction of our model. Examples are taken from the UNC validation set.

4.4 Related Work

In this section, we review related work in several related areas: Semantic segmentation classifies the object category of each pixel in an image without language input. Referring expression comprehension locates a bounding box for the object(s) described in the language input. Image segmentation from referring expressions generates a segmentation mask for the object(s) described in the language input. We also cover work on language-conditional (dynamic) filters and studies that use them to modulate deep-learning models with language.

4.4.1 Semantic Segmentation

Primitive semantic segmentation models are based on Fully Convolutional Networks (FCN) [Long et al., 2015]. DeepLab [Chen et al., 2017b] and U-Net [Ronneberger et al., 2015] are the most notable state-of-the-art semantic segmentation models related to our work. DeepLab replaces regular convolutions with atrous (dilated) convolutions in the last residual block of ResNets [He et al., 2016] and implements Atrous Spatial Pyramid Pooling (ASPP) which fuses multi-scale visual information. The U-Net architecture [Ronneberger et al., 2015] improves over the standard FCN by connecting contracting (bottom-up) and expanding (top-down) paths at the same resolution: the output of the encoder layer at each level is passed to the decoder at the same level.

4.4.2 Referring Expression Comprehension

Early models for this task were typically built using a hybrid LSTM-CNN architecture [Hu et al., 2016b, Mao et al., 2016]. Newer models [Hu et al., 2017c, Yu et al., 2016, Yu et al., 2018b, Wang et al., 2019] use a Region-based CNN (R-CNN) variant [Girshick et al., 2014, Ren et al., 2017, He et al., 2017] as a sub-component to generate object proposals. Among the literature, Compositional Modular Network (CMN) [Hu et al., 2017c], Modular Attention Network (MAttNet) [Yu et al., 2018b] and Neural Module Tree Networks (NMTree) [Liu et al., 2019a] are the most notable state-of-the-art methods. CMN implements neural modules [Andreas et al., 2016b] to generate subject/object/relationship

score triplets for the given description and the image regions proposed by Faster R-CNN [Ren et al., 2017]. MAttNet improves CMN by adding visual subject-object-relationship modules and integrating an attention mechanism to subject-object-relationship modules. NMTree generates dependency parsing tree of the given textual input, assigns neural modules for nodes, then starting from bottom, it tries to predict the most relevant bounding box for all nodes recurrently. At the end, it gives a final prediction which is for the root node.

4.4.3 Image Segmentation from Referring Expressions

Since the goal is predicting the segmentation mask of the referred objects, several approaches have been proposed in this field to incorporate the textual and visual features. RRN citeli2018referring which has a structure similar to U-Net [Ronneberger et al., 2015] is built on top of a Convolutional LSTM (ConvLSTM) [SHI et al., 2015] network. Unlike our model, ConvLSTM filters are not generated from language representation and the multi-modal representation is used only in the initial time step. DMN [Margffoy-Tuay et al., 2018] generates 1×1 language-conditional filters for language representation of each word. It performs convolution operation on visual representation with language-conditional filters to generate multi-modal representation for each word. Like RMI [Liu et al., 2017], word-level multi-modal representations are fed as input to a multi-modal recurrent network to obtain multi-modal representation for image/language pairs. CAC [Chen et al., 2019c] also generates 1×1 language-conditional dynamic filters. Unlike our model, CAC applies these dynamic filters to single resolution / single feature map and additionally generates location-specific dynamic filters (e.g. left, bottom) to capture relations between the objects exist at the different parts of the image. CMSA [Ye et al., 2019] obtains a visual-textual co-embedding by appending the embedding of each word to the visual feature map, then it applies a series of self-attention to predict the segmentation mask. Step-ConvRNN [Chen et al., 2019a] also combines the top-down visual features with textual features to obtain a co-embedding and uses a ConvRNN to iteratively refine a heatmap for image segmentation. BRINet [Hu et al., 2020] calculates a multi-modal feature for each word then using the dependency parse tree of the referring expressing, updates the representation of each word

using a graph convolution network. extends Step-ConvRNN using the vision to language and language to vision attentions. LSCM [Hui et al., 2020] uses dependency parsing trees of referring expressions to softly integrate the word specific multi-modal features and replicates this process for different level of visual features. It uses a ConvLSTM to fuse the outputs for the different levels. Since the average number of words in the expressions in the mentioned datasets is too small, withing the network, having a visual feature map for each word or concatenating the word representation with the feature maps (e.g., CMSA, Step-ConvRNN, BRINet, LSCM) are feasible. However, it is not scalable to the long expressions because of the computational memory requirements. Unlike these models, our model uses a simpler architecture by generating a language filter for each visual processing layer and modulates the visual input with the simple convolution operation.

4.4.4 Language-Conditional Filters

To control a deep learning model with language, early work such as Modulated ResNet (MODERN) [De Vries et al., 2017] and Feature-wise Linear Modulation (FiLM) [Perez et al., 2018] used conditional batch normalization layers with only language-conditioned coefficients rather than customized filters. [Li et al., 2017] is the first work which generates dynamic language-conditional filters. [Gao et al., 2018] proposes a VQA solution method which has a group convolutional layer whose filters generated from the question input. [Gavrilyuk et al., 2018] introduces a new task called actor and action segmentation and to solve this task, proposes an architecture which uses dynamic filters for multiple resolutions. Similar to our work, [Misra et al., 2018] adds language conditional filters to a U-Net based architecture for the task of mapping instructions to actions in virtual environments. It only modulates top-down visual processing with language.

Referring expression models that incorporate language-conditional filters into the architecture include [Chen et al., 2019c, Margffoy-Tuay et al., 2018]. [Margffoy-Tuay et al., 2018] generates language-conditional filters for words individually rather than whole sentence. [Chen et al., 2019c] generates 1 x 1 language-conditional filters from expressions. To make 1 x 1 language-conditional filters spatially aware, different filters are generated for different image regions (e.g. top, left, right, bottom). Our main contribution in this work

is an explicit evaluation of language conditional filters for both bottom-up and top-down visual processing in comparison to only using language for top-down attention control.

4.5 Conclusions

In this chapter, we showed that modulating not only top-down but also bottom-up visual processing with language conditioned kernels improves the language grounding performance of a model. We showed the effectiveness of our approach on 4 different referring expression understanding benchmarks. Our model, which is a variant of U-Net architecture with language conditioned branches, achieves state-of-the-art or comparable results on the benchmark datasets. Our proposed model performs significantly (≈ 6 IoU) better than a baseline which uses language to only direct top-down attention. Our future work will focus on experimenting with the proposed model on different language-vision problems such as visual question answering and using it as a sub-component to solve a far more challenging task like mapping natural language instructions to sequences of actions.

Chapter 5

CONCLUSION & FUTURE DIRECTIONS

In this thesis, we presented a series of cognitively-inspired neural networks for grounded language understanding. We showed that well-designed neural networks that make use of the inductive biases inspired from the human cognition, language compositionality and task patterns can achieve better accuracy and generalization. We also demonstrated how synthetic data generation could be helpful to further examine the models in terms of the performance on the different sub-tasks.

In Chapter 2, we presented a state-of-the-art encoder-decoder network with a channel-based perceptual attention mechanism to benefit from the language priors for the navigational instruction following task. We also proposed an improved world representation to allow the model spatially reason over the perception. Further, we analyzed the available dataset regarding its size and balance, and demonstrated that performance on a small, fixed-size dataset is no longer a good measure to differentiate state-of-the-art models. To overcome the scarce and unbalanced data problem, we presented a synthetic data generation framework and an evaluation metric which makes use of the unlimited data.

In Chapter 3, we presented a real robot system that understands natural language instructions. We demonstrated the usage of the Neural Module Networks approach in a real robot by training the modules on a simulated data and learning the world representation separately. Further, we proposed a Bayesian learning approach to maintain a common ground between a user and a robot to reason about the instruction. We showed how we can utilize the implicit information in the instruction to update the perceptual belief. We also demonstrated the compositionality of the Neural Module Networks approach with a simulated language game.

In both chapters, we demonstrated systems that work on the high-level perceptual representation and use language as a high-level guide. In addition to this, in Chapter 4,

we showed modulating both “*bottom-up*” (low-level) and “*top-down*” (high-level) visual processing with language significantly improves the language grounding performance. Our model which allows us to modulate both bottom-up and top-down visual processing with language explicitly achieved state-of-the-art or comparable results on four different image segmentation from referring expressions datasets.

5.1 Future Directions

In this section, we discuss some of the potential research directions that can further extend the grounded language understanding domain.

5.1.1 Learning to Simulate the World

Although we worked on the grounded language learning tasks in this thesis, text based language understanding also requires language grounding. [Lakoff and Johnson, 1980, Lakoff et al., 1999] proposed that embodied and situated knowledge provides grounding for abstract concepts like metaphors. Their argument was that people have a multitude of experiences about their bodies (e.g., running) and situations (e.g., vertically), and abstract concepts build on that knowledge metaphorically. For example, increase in the stock prices for a period can be understood as a run (“bull run”), and emotions that we experience can be understood as vertically (“mood is up or down”). People extensively use such metaphors to communicate their ideas [Turner, 1996]. Similarly, [Boroditsky and Ramscar, 2002, Gibbs Jr, 2005, Barsalou and Wiemer-Hastings, 2005] show that understanding metaphors through simulation plays an important role in thought.

Despite the current natural language understanding (NLU) systems have shown great success on mainstream text only tasks [Devlin et al., 2019, Liu et al., 2019b], there are significant differences between these systems and human-like language understanding in terms of generalization, grounding, reasoning, and explainability [Glockner et al., 2018, McCoy et al., 2019a, McCoy et al., 2019b, Nie et al., 2019, Yogatama et al., 2019]. For example, [Bisk et al., 2020] showed that a near state-of-the-art model performs poorly on a text-based question answering task which requires understanding of physical actions.

This demonstrates that these systems lack of the ability to simulate actions and their effects.

One approach for the simulation of the world is *learning world models*. [Ha and Schmidhuber, 2018] proposed a recurrent neural network that predicts next states (as frames) of the environment given the past experiences. [Kipf et al., 2019] proposed a method to learn the abstract representation of objects and actions in a graph neural network setting to model the effects of the actions on the objects. [Hafner et al., 2019] showed a framework where an agent plans its actions in a learned world model. However, language and learned world models have not been linked in these studies. Learning world models and reasoning over the learned world models for language understanding would bring the NLU systems closer to the level of human-like language understanding.

Learning Intuitive physics is another direction to incorporate reasoning over the physical world in the embodied studies. *Intuitive physics* is a crucial component of our cognition, especially it performs an important role in building our commonsense knowledge. We understand how the objects move, what the rigid body is, how the liquids behave if we pour them in a container or how an object inside a box gets affected if we move the box based on the intuitive physics. As young infants, we quickly start to comprehend how the world around us is working. Infants expect that the object smoothly move along in a path at ages of 2 months [Spelke, 1990, Spelke et al., 1995]. Around 6 months, infants start to distinguish the soft bodies, rigid bodies and liquids between them [Hespos et al., 2009]. At year 1, infants comprehend the basic physical concepts such as inertia, collisions, containment and support [Baillargeon, 2004, Hespos and Baillargeon, 2008, Baillargeon et al., 2009].

Recently, learning models reasoning about intuitive physics gained attention from the research community. [Mottaghi et al., 2016] proposed a deep learning model for predicting the dynamics of objects in static images. [Lerer et al., 2016] presented PhysNet, a deep learning model, for the prediction of the stability of the block towers in a simulated environment. [Janner et al., 2018] proposed a model to predict the next state of the world. Their model jointly learns an abstract object representations from an image, a function to predict the configuration of objects evolved over time and a rendering function to map the predicted configuration as image. Despite these models providing reasoning for the

intuitive physics, the language interaction is missing. Recently, [Yi et al., 2019] introduced the CLEVRER dataset for visual question answering that requires wide range of physical reasoning. The dataset includes different type of questions such as descriptive (e.g., ‘what color’), explanatory (‘what’s responsible for’), predictive (‘what will happen next’), and counterfactual (‘what if’). [Ates et al., 2020] developed the CRAFT dataset which also introduces cause, enable, and prevent related questions. [Yi et al., 2019] showed that the state-of-the-art visual question answering models [Johnson et al., 2017b, Hudson and Manning, 2018] fail to solve CLEVRER, which demonstrates the need for the further research on developing models that are able to understand intuitive physics and reason over the world to ground the language inputs.

5.1.2 Language as Reward

In this thesis, we presented systems for the tasks that have weakly-supervised signal (i.e., correct path for navigational instruction following, ground-truth objects for robotic instruction following, and ground-truth segmentation masks for referring expression understanding) available for language grounding. However, acquiring the training data for language grounding problems is a grand challenge. We use language in a complex environment. To have free form natural sentences, we need to have environments complex enough to provide rich relations. Hence, preparing both environment and language parts of data is labor intensive work. One common approach to tackle this problem is using Reinforcement Learning techniques where an agent gets positive or negative feedback from the environment [Misra et al., 2017, Yu et al., 2017, Chaplot et al., 2017, Hermann et al., 2017, Misra et al., 2018, Chen et al., 2019b]. However, these approaches require a large amount of examples to learn from feedback due to the credit assignment problem [Minsky, 1961], where the determining the actions that lead positive or negative outcomes is problematic. It is also expensive to develop an environment that provides feedback. In this cases, the work we presented in Chapter 3 can be extended to provide human feedback to an agent for learning the task relevant behaviours. Human could provide binary reward for the evaluation of whether the agent reached the goal or not, e.g., ”you are not at the target position”. Even, a human user could provide richer feedback about where the agent

failed, e.g., "you turned left instead of right at the previous intersection", which would ease off the credit assignment problem.

5.1.3 Grounded Language Learning in the Social World

In this thesis, we demonstrated systems that learn to ground language to perceptual representations to execute the linguistic instruction. Although utterances find their meanings on non-linguistic world, we generate words with a communicative intent such as giving an instruction, conveying an information or simply socially engaging [Bender and Koller, 2020]. This social engagements are one of the important factors of language learning in children. [Snow et al., 1976] and [Kuhl, 2007] showed that children do not start to grasp a language from a passive experiences like watching TV, however, they start to distinguish phonemic differences after having brief interactions with others in the non-native language. Similarly, a series of language acquisition studies suggests that after the first year, children start to learn meanings of words through social cues [Hollich et al., 2000, Baldwin and Moses, 2001, Golinkoff and Hirsh-Pasek, 2006, Nelson, 2007, Brooks and Meltzoff, 2008, Tomasello, 2010]. [Baldwin and Moses, 2001] argues that interactions are not enough for language acquisition, the critical factor is joint attention where the child and a caregiver both have focus on the same thing intentionally. This argument has been supported with several experiments showing that infants have more vocabularies in cases that the caregivers follow their attention and give language labels about the context [Tomasello and Farrar, 1986]; that infants do not gather the labels when they do not see the caregiver, but pick up the labels when the caregiver is present and have the joint attention [Moore and Dunham, 1995]; that infants have larger vocabularies and a faster rate of vocabulary growth if they follow the eye-gaze of the speaker longer time than the infants lose their focus more quickly [Brooks and Meltzoff, 2008].

In conclusion, we learn language in a social world where we guess the communication intent of the another human through social context and joint attentions. Similarly, to achieve human level understanding, an intelligent robot should learn language from linguistic, perceptual and social signals while interacting with the others and the environment.

BIBLIOGRAPHY

- [Agrawal et al., 2016] Agrawal, A., Batra, D., and Parikh, D. (2016). Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960.
- [Aktan-Erciyes and Göksun, 2019] Aktan-Erciyes, A. and Göksun, T. (2019). Early event understanding predicts later verb comprehension and motion event lexicalization. *Developmental psychology*, 55(11):2249.
- [Alchourrón et al., 1985] Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of symbolic logic*, pages 510–530.
- [Anderson et al., 2018a] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (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.
- [Anderson et al., 2018b] Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., and van den Hengel, A. (2018b). Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2.
- [Andreas and Klein, 2015] Andreas, J. and Klein, D. (2015). Alignment-based compositional semantics for instruction following. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1165–1174.
- [Andreas et al., 2016a] Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2016a).

Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*.

[Andreas et al., 2016b] Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2016b). Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

[Antol et al., 2015] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433.

[Artzi et al., 2014] Artzi, Y., Das, D., and Petrov, S. (2014). Learning compact lexicons for ccg semantic parsing. In *EMNLP*, pages 1273–1283.

[Artzi and Zettlemoyer, 2013] Artzi, Y. and Zettlemoyer, L. (2013). Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.

[Ates et al., 2020] Ates, T., Atesoglu, M. S., Yigit, C., Kesen, I., Kobas, M., Erdem, E., Erdem, A., Goksun, T., and Yuret, D. (2020). Craft: A benchmark for causal reasoning about forces and interactions. *arXiv preprint arXiv:2012.04293*.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Baillargeon, 2004] Baillargeon, R. (2004). Infants’ physical world. *Current directions in psychological science*, 13(3):89–94.

[Baillargeon et al., 2009] Baillargeon, R., Li, J., Ng, W., and Yuan, S. (2009). An account of infants’ physical reasoning. *Learning and the infant mind*, pages 66–116.

[Baker et al., 1998] Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association*

for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1, pages 86–90. Association for Computational Linguistics.

- [Baldwin and Moses, 2001] Baldwin, D. A. and Moses, L. J. (2001). Links between social understanding and early word learning: Challenges to current accounts. *Social Development*, 10(3):309–329.
- [Barsalou, 2008] Barsalou, L. W. (2008). Grounded cognition. *Annu. Rev. Psychol.*, 59:617–645.
- [Barsalou and Wiemer-Hastings, 2005] Barsalou, L. W. and Wiemer-Hastings, K. (2005). Situating abstract concepts. *Grounding cognition: The role of perception and action in memory, language, and thought*, pages 129–163.
- [Ben-Younes et al., 2017] Ben-Younes, H., Cadene, R., Cord, M., and Thome, N. (2017). Mutan: Multimodal tucker fusion for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3.
- [Bender and Koller, 2020] Bender, E. M. and Koller, A. (2020). Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.
- [Bisk et al., 2020] Bisk, Y., Zellers, R., LeBras, R., Gao, J., and Choi, Y. (2020). Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, pages 7432–7439.
- [Bloom, 2002] Bloom, P. (2002). *How children learn the meanings of words*. MIT press.
- [Borghi et al., 2004] Borghi, A. M., Glenberg, A. M., and Kaschak, M. P. (2004). Putting words in perspective. *Memory & Cognition*, 32(6):863–873.
- [Bornstein et al., 2004] Bornstein, M. H., Cote, L. R., Maital, S., Painter, K., Park, S.-Y., Pascual, L., Pêcheux, M.-G., Ruel, J., Venuti, P., and Vyt, A. (2004). Cross-linguistic

analysis of vocabulary in young children: Spanish, dutch, french, hebrew, italian, korean, and american english. *Child development*, 75(4):1115–1139.

[Boroditsky and Ramscar, 2002] Boroditsky, L. and Ramscar, M. (2002). The roles of body and mind in abstract thought. *Psychological science*, 13(2):185–189.

[Börschinger et al., 2011] Börschinger, B., Jones, B. K., and Johnson, M. (2011). Reducing grounded learning tasks to grammatical inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1416–1425. Association for Computational Linguistics.

[Boutonnet et al., 2013] Boutonnet, B., Dering, B., Viñas-Guasch, N., and Thierry, G. (2013). Seeing objects through the language glass. *Journal of Cognitive Neuroscience*, 25(10):1702–1710.

[Boutonnet and Lupyan, 2015] Boutonnet, B. and Lupyan, G. (2015). Words jump-start vision: A label advantage in object recognition. *Journal of Neuroscience*, 35(25):9329–9335.

[Bowerman and Levinson,] Bowerman, M. and Levinson, S. 2001: Language acquisition and conceptual development.

[Brooks and Meltzoff, 2008] Brooks, R. and Meltzoff, A. N. (2008). Infant gaze following and pointing predict accelerated vocabulary growth through two years of age: A longitudinal, growth curve modeling study. *Journal of child language*, 35(1):207.

[Campeau-Lecours et al., 2019] Campeau-Lecours, A., Lamontagne, H., Latour, S., Fautoux, P., Maheu, V., Boucher, F., Deguire, C., and L’Ecuyer, L.-J. C. (2019). Kinova modular robot arms for service robotics applications. In *Rapid Automation: Concepts, Methodologies, Tools, and Applications*, pages 693–719. IGI Global.

[Chang et al., 2017] Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. (2017). Matterport3d: Learning from rgb-d data

- in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE.
- [Chaplot et al., 2017] Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., and Salakhutdinov, R. (2017). Gated-attention architectures for task-oriented language grounding. *arXiv preprint arXiv:1706.07230*.
- [Chen et al., 2019a] Chen, D.-J., Jia, S., Lo, Y.-C., Chen, H.-T., and Liu, T.-L. (2019a). See-through-text grouping for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7454–7463.
- [Chen, 2012] Chen, D. L. (2012). Fast online lexicon learning for grounded language acquisition. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 430–439.
- [Chen and Mooney, 2011] Chen, D. L. and Mooney, R. J. (2011). Learning to interpret natural language navigation instructions from observations. In *AAAI*, volume 2, pages 1–2.
- [Chen et al., 2016] Chen, H., Qi, X. J., Cheng, J. Z., and Heng, P. A. (2016). Deep contextual networks for neuronal structure segmentation. In *Thirtieth AAAI conference on artificial intelligence*.
- [Chen et al., 2019b] Chen, H., Suhr, A., Misra, D., Snavely, N., and Artzi, Y. (2019b). Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.
- [Chen et al., 2017a] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., and Chua, T.-S. (2017a). Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6298–6306. IEEE.

- [Chen et al., 2017b] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017b). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.
- [Chen et al., 2019c] Chen, Y.-W., Tsai, Y.-H., Wang, T., Lin, Y.-Y., and Yang, M.-H. (2019c). Referring expression object segmentation with caption-aware consistency. In *British Machine Vision Conference (BMVC)*.
- [Choi and Bowerman, 1991] Choi, S. and Bowerman, M. (1991). Learning to express motion events in english and korean: The influence of language-specific lexicalization patterns. *Cognition*, 41(1-3):83–121.
- [Cirik et al., 2018] Cirik, V., Berg-Kirkpatrick, T., and Morency, L.-P. (2018). Using syntax to ground referring expressions in natural images. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [Connor et al., 2004] Connor, C. E., Egeth, H. E., and Yantis, S. (2004). Visual attention: bottom-up versus top-down. *Current biology*, 14(19):R850–R852.
- [Coradeschi and Saffiotti, 2000] Coradeschi, S. and Saffiotti, A. (2000). Anchoring symbols to sensor data: preliminary report. In *Proc. of the 17th AAAI Conf.*, pages 129–135, Menlo Park, CA. AAAI Press. Online at <http://www.aass.oru.se/~asaffio/>.
- [Coradeschi and Saffiotti, 2003] Coradeschi, S. and Saffiotti, A. (2003). An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2):85–96.
- [Corbetta and Shulman, 2002] Corbetta, M. and Shulman, G. L. (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience*, 3(3):201–215.
- [Das et al., 2018] Das, A., Gkioxari, G., Lee, S., Parikh, D., and Batra, D. (2018). Neural modular control for embodied question answering. *arXiv preprint arXiv:1810.11181*.

- [Das et al., 2016] Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M., Parikh, D., and Batra, D. (2016). Visual dialog. *arXiv preprint arXiv:1611.08669*.
- [De Vries et al., 2017] De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. (2017). Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604.
- [Dessalegn and Landau, 2008] Dessalegn, B. and Landau, B. (2008). More than meets the eye: The role of language in binding and maintaining feature conjunctions. *Psychological science*, 19(2):189–195.
- [Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.
- [Dils and Boroditsky, 2010] Dils, A. T. and Boroditsky, L. (2010). Processing unrelated language can change what you see. *Psychonomic bulletin & review*, 17(6):882–888.
- [Dou et al., 2017] Dou, Q., Yu, L., Chen, H., Jin, Y., Yang, X., Qin, J., and Heng, P. (2017). 3d deeply supervised network for automated segmentation of volumetric medical images. *Medical image analysis*, 41:40.
- [Dumitru et al., 2013] Dumitru, M. L., Joergensen, G. H., Cruickshank, A. G., and Altmann, G. T. (2013). Language-guided visual processing affects reasoning: The role of referential and spatial anchoring. *Consciousness and Cognition*, 22(2):562–571.
- [Elfring et al., 2013] Elfring, J., van den Dries, S., van de Molengraft, M., and Steinbuch, M. (2013). Semantic world modeling using probabilistic multiple hypothesis anchoring. *Robotics and Autonomous Systems*, 61(2):95–105.
- [Escalante et al., 2010] Escalante, H. J., Hernández, C. A., Gonzalez, J. A., López-López, A., Montes, M., Morales, E. F., Sucar, L. E., Villaseñor, L., and Grubinger, M. (2010).

The segmented and annotated iapr tc-12 benchmark. *Computer vision and image understanding*, 114(4):419–428.

[Evans, 2009] Evans, V. (2009). *How words mean: Lexical concepts, cognitive models, and meaning construction*. Oxford University Press on Demand.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

[Fodor, 1975] Fodor, J. A. (1975). *The language of thought*, volume 5. Harvard university press.

[Fodor, 1983] Fodor, J. A. (1983). *The modularity of mind*. MIT press.

[Forder and Lupyan, 2019] Forder, L. and Lupyan, G. (2019). Hearing words changes color perception: Facilitation of color discrimination by verbal and visual cues. *Journal of Experimental Psychology: General*, 148(7):1105.

[Fried et al., 2018] Fried, D., Andreas, J., and Klein, D. (2018). Unified pragmatic models for generating and following instructions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1951–1963.

[Gallese and Lakoff, 2005] Gallese, V. and Lakoff, G. (2005). The brain’s concepts: The role of the sensory-motor system in conceptual knowledge. *Cognitive neuropsychology*, 22(3-4):455–479.

[Gao et al., 2018] Gao, P., Li, H., Li, S., Lu, P., Li, Y., Hoi, S. C., and Wang, X. (2018). Question-guided hybrid convolution for visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 469–485.

[Gärdenfors, 1988] Gärdenfors, P. (1988). *Knowledge in flux: Modeling the dynamics of epistemic states*. The MIT press.

- [Gärdenfors, 1990] Gärdenfors, P. (1990). The dynamics of belief systems: Foundations vs. coherence theories. *Revue internationale de philosophie*, pages 24–46.
- [Gärdenfors, 2004] Gärdenfors, P. (2004). *Conceptual spaces: The geometry of thought*. MIT press.
- [Gardenfors, 2014] Gardenfors, P. (2014). *The geometry of meaning: Semantics based on conceptual spaces*. MIT press.
- [Gavrilyuk et al., 2018] Gavrilyuk, K., Ghodrati, A., Li, Z., and Snoek, C. G. (2018). Actor and action video segmentation from a sentence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5958–5966.
- [Gibbs Jr, 2005] Gibbs Jr, R. W. (2005). *Embodiment and cognitive science*. Cambridge University Press.
- [Girshick, 2015] Girshick, R. (2015). Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*.
- [Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*.
- [Glockner et al., 2018] Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655.
- [Goertzel et al., 2010] Goertzel, B., Pennachin, C., Araujo, S., Silva, F., Queiroz, M., Lian, R., Silva, W., Ross, M., Vepstas, L., and Senna, A. (2010). A general intelligence oriented architecture for embodied natural language processing. In *Proceedings of the 3d Conference on Artificial General Intelligence*.

- [Göksun et al., 2017] Göksun, T., Aktan-Erciyes, A., Hirsh-Pasek, K., and Golinkoff, R. M. (2017). Event perception and language learning. *Social environment and cognition in language development: Studies in honor of Ayhan Aksu-Koç*, 21:179.
- [Golinkoff and Hirsh-Pasek, 2006] Golinkoff, R. M. and Hirsh-Pasek, K. (2006). Baby wordsmith: From associationist to social sophisticate. *Current directions in psychological science*, 15(1):30–33.
- [Goyal et al., 2017] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (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.
- [Graves and Schmidhuber, 2005] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.
- [Ha and Schmidhuber, 2018] Ha, D. and Schmidhuber, J. (2018). Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pages 2450–2462.
- [Hafner et al., 2019] Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*.
- [Harnad, 1990] Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.
- [Hart et al., 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

- [Hawkins et al., 2018] Hawkins, R. X., Franke, M., Smith, K., and Goodman, N. D. (2018). Emerging abstractions: Lexical conventions are shaped by communicative context. In *CogSci*.
- [He et al., 2017] He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Hermann et al., 2017] Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W. M., Jaderberg, M., Teplyashin, D., et al. (2017). Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*.
- [Hespos and Baillargeon, 2008] Hespos, S. J. and Baillargeon, R. (2008). Young infants’ actions reveal their developing knowledge of support variables: Converging evidence for violation-of-expectation findings. *Cognition*, 107(1):304–316.
- [Hespos et al., 2009] Hespos, S. J., Ferry, A. L., and Rips, L. J. (2009). Five-month-old infants have different expectations for solids and liquids. *Psychological Science*, 20(5):603–611.
- [Hochreiter and Schmidhuber, 1997a] Hochreiter, S. and Schmidhuber, J. (1997a). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Hochreiter and Schmidhuber, 1997b] Hochreiter, S. and Schmidhuber, J. (1997b). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Hollich et al., 2000] Hollich, G., Hirsh-Pasek, K., and Golinkoff, R. (2000). Breaking the language barrier: An emergentist coalition model for the origins of word learning. society for research in child development monograph series. *Serial no. 262*, 65(3).

- [Hu et al., 2017a] Hu, R., Andreas, J., Rohrbach, M., Darrell, T., and Saenko, K. (2017a). Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813.
- [Hu et al., 2017b] Hu, R., Rohrbach, M., Andreas, J., Darrell, T., and Saenko, K. (2017b). Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1115–1124.
- [Hu et al., 2017c] Hu, R., Rohrbach, M., Andreas, J., Darrell, T., and Saenko, K. (2017c). Modeling relationships in referential expressions with compositional modular networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Hu et al., 2016a] Hu, R., Rohrbach, M., and Darrell, T. (2016a). Segmentation from natural language expressions. *Lecture Notes in Computer Science*, page 108–124.
- [Hu et al., 2016b] Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., and Darrell, T. (2016b). Natural language object retrieval. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Hu et al., 2020] Hu, Z., Feng, G., Sun, J., Zhang, L., and Lu, H. (2020). Bi-directional relationship inferring network for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4424–4433.
- [Hudson and Manning, 2018] Hudson, D. A. and Manning, C. D. (2018). Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*.
- [Hui et al., 2020] Hui, T., Liu, S., Huang, S., Li, G., Yu, S., Zhang, F., and Han, J. (2020). Linguistic structure guided context modeling for referring image segmentation.
- [Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

- [Jackendoff, 2002] Jackendoff, R. (2002). *Foundations of language: Brain, meaning, grammar, evolution*. Oxford University Press, USA.
- [Janner et al., 2018] Janner, M., Levine, S., Freeman, W. T., Tenenbaum, J. B., Finn, C., and Wu, J. (2018). Reasoning about physical interactions with object-oriented prediction and planning. In *International Conference on Learning Representations*.
- [Johnson et al., 2016] Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. (2016). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *arXiv preprint arXiv:1612.06890*.
- [Johnson et al., 2017a] Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. (2017a). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE.
- [Johnson et al., 2017b] Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. (2017b). Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2989–2998.
- [Kant, 1878] Kant, I. (1878). *Prolegomena zu einer jeden Künftigen Metaphysik: die als Wissenschaft wird auftreten können*. Verlag von Leopold Voss.
- [Kate and Mooney, 2006] Kate, R. J. and Mooney, R. J. (2006). Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 913–920. Association for Computational Linguistics.
- [Kazemzadeh et al., 2014] Kazemzadeh, S., Ordonez, V., Matten, M., and Berg, T. (2014). Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

- [Kempka et al., 2016] Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Jaśkowski, W. (2016). Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, pages 1–8. IEEE.
- [Kiefer, 1953] Kiefer, J. (1953). Sequential minimax search for a maximum. *Proceedings of the American mathematical society*, 4(3):502–506.
- [Kiela et al., 2016] Kiela, D., Bulat, L., Vero, A. L., and Clark, S. (2016). Virtual embodiment: A scalable long-term strategy for artificial intelligence research. *arXiv preprint arXiv:1610.07432*.
- [Kim and Mooney, 2012] Kim, J. and Mooney, R. J. (2012). Unsupervised pcfg induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 433–444. Association for Computational Linguistics.
- [Kim and Mooney, 2013] Kim, J. and Mooney, R. J. (2013). Adapting discriminative reranking to grounded language learning. In *ACL (1)*, pages 218–227.
- [Kim et al., 2016] Kim, J.-H., Lee, S.-W., Kwak, D., Heo, M.-O., Kim, J., Ha, J.-W., and Zhang, B.-T. (2016). Multimodal residual learning for visual qa. In *Advances in neural information processing systems*, pages 361–369.
- [Kingma and Ba, 2014a] Kingma, D. and Ba, J. (2014a). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Kingma and Ba, 2014b] Kingma, D. P. and Ba, J. (2014b). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Kintsch and Van Dijk, 1978] Kintsch, W. and Van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological review*, 85(5):363.

- [Kipf et al., 2019] Kipf, T., van der Pol, E., and Welling, M. (2019). Contrastive learning of structured world models. In *International Conference on Learning Representations*.
- [Kočiský et al., 2016] Kočiský, T., Melis, G., Grefenstette, E., Dyer, C., Ling, W., Blunsom, P., and Hermann, K. M. (2016). Semantic parsing with semi-supervised sequential autoencoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1078–1087.
- [Kollar et al., 2013] Kollar, T., Tellex, S., Walter, M. R., Huang, A., Bachrach, A., Hemachandra, S., Brunskill, E., Banerjee, A., Roy, D., Teller, S., et al. (2013). Generalized grounding graphs: A probabilistic framework for understanding grounded language. *JAIR*.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- [Kuhl, 2007] Kuhl, P. K. (2007). Is speech learning ‘gated’ by the social brain? *Developmental science*, 10(1):110–120.
- [Kuhnle and Copestake, 2017] Kuhnle, A. and Copestake, A. (2017). Shapeworld—a new test methodology for multimodal language understanding. *arXiv preprint arXiv:1704.04517*.
- [Kuhnle and Copestake, 2018] Kuhnle, A. and Copestake, A. (2018). Deep learning evaluation using deep linguistic processing. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 17–23.
- [Lakoff and Johnson, 1980] Lakoff, G. and Johnson, M. (1980). *Metaphor we live by*. Chicago/London.
- [Lakoff et al., 1999] Lakoff, G., Johnson, M., et al. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*, volume 640. Basic books New York.

- [LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.
- [Lee et al., 2015] Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. (2015). Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570.
- [Lemaignan et al., 2011] Lemaignan, S., Ros, R., Alami, R., and Beetz, M. (2011). What are you talking about? grounding dialogue in a perspective-aware robotic architecture. In *2011 RO-MAN*, pages 107–112. IEEE.
- [Lemaignan et al., 2012] Lemaignan, S., Ros, R., Sisbot, E. A., Alami, R., and Beetz, M. (2012). Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *International Journal of Social Robotics*, 4(2):181–199.
- [Lerer et al., 2016] Lerer, A., Gross, S., and Fergus, R. (2016). Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*.
- [Li et al., 2018] Li, R., Li, K., Kuo, Y.-C., Shu, M., Qi, X., Shen, X., and Jia, J. (2018). Referring image segmentation via recurrent refinement networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753.
- [Li et al., 2017] Li, Z., Tao, R., Gavves, E., Snoek, C. G., and Smeulders, A. W. (2017). Tracking by natural language specification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6495–6503.
- [Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- [Liu et al., 2017] Liu, C., Lin, Z. L., Shen, X., Yang, J., Lu, X., and Yuille, A. L. (2017). Recurrent multimodal interaction for referring image segmentation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1280–1289.

- [Liu et al., 2019a] Liu, D., Zhang, H., Wu, F., and Zha, Z.-J. (2019a). Learning to assemble neural module tree networks for visual grounding. In *The IEEE International Conference on Computer Vision (ICCV)*.
- [Liu and Singh, 2004] Liu, H. and Singh, P. (2004). Conceptnet—a practical common-sense reasoning tool-kit. *BT technology journal*, 22(4):211–226.
- [Liu et al., 2019b] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- [Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.
- [Loutfi et al., 2005] Loutfi, A., Coradeschi, S., and Saffiotti, A. (2005). Maintaining coherent perceptual information using anchoring. In *Proc. of the 19th IJCAI Conf.*, pages 1477–1482, Edinburgh, UK.
- [Lu et al., 2017] Lu, J., Xiong, C., Parikh, D., and Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3242–3250. IEEE.
- [Lu et al., 2016] Lu, J., Yang, J., Batra, D., and Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pages 289–297.
- [Lupyan and Clark, 2015] Lupyan, G. and Clark, A. (2015). Words and the world: Predictive coding and the language-perception-cognition interface. *Current Directions in Psychological Science*, 24(4):279–284.

- [Lupyan and Swingley, 2012] Lupyan, G. and Swingley, D. (2012). Self-directed speech affects visual search performance. *Quarterly Journal of Experimental Psychology*, 65(6):1068–1085.
- [Lupyan and Ward, 2013] Lupyan, G. and Ward, E. J. (2013). Language can boost otherwise unseen objects into visual awareness. *Proceedings of the National Academy of Sciences*, 110(35):14196–14201.
- [Maas et al., 2013] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.
- [Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- [MacMahon et al., 2006] MacMahon, M., Stankiewicz, B., and Kuipers, B. (2006). Walk the talk: Connecting language, knowledge, and action in route instructions. *Def*, 2(6):4.
- [Malinowski et al., 2015] Malinowski, M., Rohrbach, M., and Fritz, M. (2015). Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE international conference on computer vision*, pages 1–9.
- [Mao et al., 2016] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., and Murphy, K. (2016). Generation and comprehension of unambiguous object descriptions. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Margffoy-Tuay et al., 2018] Margffoy-Tuay, E., Pérez, J. C., Botero, E., and Arbeláez, P. (2018). Dynamic multimodal instance segmentation guided by natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 630–645.
- [Mast et al., 2016] Mast, V., Falomir, Z., and Wolter, D. (2016). Probabilistic reference and grounding with pragr for dialogues with robots. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(5):889–911.

- [Matuszek et al., 2012] Matuszek, C., FitzGerald, N., Zettlemoyer, L., Bo, L., and Fox, D. (2012). A joint model of language and perception for grounded attribute learning. *arXiv preprint arXiv:1206.6423*.
- [Matuszek et al., 2010] Matuszek, C., Fox, D., and Koscher, K. (2010). Following directions using statistical machine translation. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258. IEEE.
- [Mavridis and Roy, 2005] Mavridis, N. and Roy, D. (2005). Grounded situation models for robots: Bridging language, perception, and action. In *AAAI-05 workshop on modular construction of human-like intelligence*.
- [McCoy et al., 2019a] McCoy, R. T., Min, J., and Linzen, T. (2019a). Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.
- [McCoy et al., 2019b] McCoy, T., Pavlick, E., and Linzen, T. (2019b). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.
- [Mei et al., 2016] Mei, H., Bansal, M., and Walter, M. R. (2016). Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *AAAI*, volume 1, page 2.
- [Meteyard et al., 2007] Meteyard, L., Bahrami, B., and Vigliocco, G. (2007). Motion detection and motion verbs: Language affects low-level visual perception. *Psychological Science*, 18(11):1007–1013.
- [Mikolov et al., 2010a] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010a). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

- [Mikolov et al., 2010b] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., and Khudanpur, S. (2010b). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.
- [Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- [Minsky, 1961] Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.
- [Misra et al., 2018] Misra, D., Bennett, A., Blukis, V., Niklasson, E., Shatkhin, M., and Artzi, Y. (2018). Mapping instructions to actions in 3d environments with visual goal prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2667–2678.
- [Misra et al., 2017] Misra, D., Langford, J., and Artzi, Y. (2017). Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015.
- [Misra et al., 2014] Misra, D. K., Sung, J., Lee, K., and Saxena, A. (2014). Tell me dave: Contextsensitive grounding of natural language to mobile manipulation instructions. In *in RSS*. Citeseer.
- [Mnih et al., 2014] Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212.
- [Moore and Dunham, 1995] Moore, C. E. and Dunham, P. J. (1995). Joint attention: Its origins and role in development.
- [Mottaghi et al., 2016] Mottaghi, R., Bagherinezhad, H., Rastegari, M., and Farhadi, A. (2016). Newtonian scene understanding: Unfolding the dynamics of objects in static

- images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3521–3529.
- [Narayanan, 1999] Narayanan, S. (1999). Moving right along: A computational model of metaphoric reasoning about events. *Aaai/iaai*, 121127.
- [Nelson, 2007] Nelson, K. (2007). Becoming a language user. *Socioemotional development in the toddler years: Transitions and transformations*, pages 221–240.
- [Nie et al., 2019] Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2019). Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.
- [Nilsson, 1984] Nilsson, N. J. (1984). Shakey the robot. Technical report, SRI International, Menlo Park CA.
- [Padmakumar et al., 2017] Padmakumar, A., Thomason, J., and Mooney, R. J. (2017). Integrated learning of dialog strategies and semantic parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- [Parish-Morris et al., 2013] Parish-Morris, J., Golinkoff, R. M., and Hirsh-Pasek, K. (2013). From coo to code: A brief story of language development.
- [Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- [Perera and Allen, 2015] Perera, I. and Allen, J. (2015). Quantity, contrast, and convention in cross-situated language comprehension. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 226–236.

- [Perez et al., 2018] Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [Persson et al., 2019] Persson, A., Zuidberg Dos Martires, P., Loutfi, A., and De Raedt, L. (2019). Semantic relational object tracking. *arXiv preprint arXiv:1902.09937*.
- [Priestley and Ward, 1994] Priestley, H. A. and Ward, M. P. (1994). A multipurpose backtracking algorithm. *Journal of Symbolic Computation*, 18(1):1–40.
- [Proffitt, 2006] Proffitt, D. R. (2006). Embodied perception and the economy of action. *Perspectives on psychological science*, 1(2):110–122.
- [Pulvermüller, 1999] Pulvermüller, F. (1999). Words in the brain’s language. *Behavioral and brain sciences*, 22(2):253–279.
- [Pulvermüller, 2013] Pulvermüller, F. (2013). How neurons make meaning: brain mechanisms for embodied and abstract-symbolic semantics. *Trends in cognitive sciences*, 17(9):458–470.
- [Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.
- [Ren et al., 2017] Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.
- [Richardson et al., 2003] Richardson, D. C., Spivey, M. J., Barsalou, L. W., and McRae, K. (2003). Spatial representations activated during real-time comprehension of verbs. *Cognitive science*, 27(5):767–780.
- [Richardson et al., 2001] Richardson, D. C., Spivey, M. J., Edelman, S., and Naples, A. J. (2001). ” language is spatial”: Experimental evidence for image schemas of concrete

- and abstract verbs. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23.
- [Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- [Roy, 2005] Roy, D. (2005). Grounding words in perception and action: computational insights. *Trends in cognitive sciences*, 9(8):389–396.
- [Rush et al., 2015] Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.
- [Saffran, 2003] Saffran, J. R. (2003). Statistical language learning: Mechanisms and constraints. *Current directions in psychological science*, 12(4):110–114.
- [Saffran et al., 1999] Saffran, J. R., Johnson, E. K., Aslin, R. N., and Newport, E. L. (1999). Statistical learning of tone sequences by human infants and adults. *Cognition*, 70(1):27–52.
- [Saint-Dizier and Viegas, 1995] Saint-Dizier, P. and Viegas, E. (1995). *Computational lexical semantics*. Cambridge University Press.
- [Samuelson et al., 2011] Samuelson, L. K., Smith, L. B., Perry, L. K., and Spencer, J. P. (2011). Grounding word learning in space. *PloS one*, 6(12):e28095.
- [Santoro et al., 2017] Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. (2017). A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976.
- [Searle, 1980] Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424.

- [She and Chai, 2017] She, L. and Chai, J. (2017). Interactive learning of grounded verb semantics towards human-robot communication. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1634–1644.
- [Shi et al., 2018] Shi, H., Li, H., Meng, F., and Wu, Q. (2018). Key-word-aware network for referring expression image segmentation. In *The European Conference on Computer Vision (ECCV)*.
- [SHI et al., 2015] SHI, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and WOO, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc.
- [Shih et al., 2016] Shih, K. J., Singh, S., and Hoiem, D. (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.
- [Shridhar et al., 2020] Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Motlaghi, R., Zettlemoyer, L., and Fox, D. (2020). Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.
- [Simmons et al., 2007] Simmons, W. K., Ramjee, V., Beauchamp, M. S., McRae, K., Martin, A., and Barsalou, L. W. (2007). A common neural substrate for perceiving and knowing about color. *Neuropsychologia*, 45(12):2802–2810.
- [Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [Siskind, 1993] Siskind, J. M. (1993). Naive physics, event perception, lexical semantics, and language acquisition. Technical report, DTIC Document.

- [Siskind, 1995] Siskind, J. M. (1995). Grounding language in perception. In *Integration of Natural Language and Vision Processing*, pages 207–227. Springer.
- [Siskind, 2001] Siskind, J. M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *J. Artif. Intell. Res.(JAIR)*, 15:31–90.
- [Slone et al., 2019] Slone, L. K., Smith, L. B., and Yu, C. (2019). Self-generated variability in object images predicts vocabulary growth. *Developmental science*, 22(6):e12816.
- [Smith and Gasser, 2005] Smith, L. and Gasser, M. (2005). The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29.
- [Snow et al., 1976] Snow, C. E., Arlman-Rupp, A., Hassing, Y., Jobse, J., Joosten, J., and Vorster, J. (1976). Mothers’ speech in three social classes. *Journal of Psycholinguistic Research*, 5(1):1–20.
- [Spelke, 1990] Spelke, E. S. (1990). Principles of object perception. *Cognitive science*, 14(1):29–56.
- [Spelke et al., 1995] Spelke, E. S., Gutheil, G., Van de Walle, G., and Osherson, D. (1995). The development of object perception. *Invitation to cognitive science, 2nd ed., Vol. 2: Visual cognition*.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- [Stanfield and Zwaan, 2001] Stanfield, R. A. and Zwaan, R. A. (2001). The effect of implied orientation derived from verbal context on picture recognition. *Psychological science*, 12(2):153–156.
- [Steedman, 2000] Steedman, M. (2000). *The syntactic process*, volume 24. MIT press Cambridge, MA.

- [Steels, 1995] Steels, L. (1995). A self-organizing spatial vocabulary. *Artificial life*, 2(3):319–332.
- [Steels and Hild, 2012] Steels, L. and Hild, M. (2012). *Language grounding in robots*. Springer Science & Business Media.
- [Stolk et al., 2016] Stolk, A., Verhagen, L., and Toni, I. (2016). Conceptual alignment: How brains achieve mutual understanding. *Trends in cognitive sciences*, 20(3):180–191.
- [Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- [Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.
- [Tellex et al., 2013] Tellex, S., Thackerll, P., Deitsl, R., Simeonovl, D., Kollar, T., and Roysl, N. (2013). Toward information theoretic human-robot dialog. *Robotics*, page 409.
- [Theeuwes, 2010] Theeuwes, J. (2010). Top–down and bottom–up control of visual selection. *Acta psychologica*, 135(2):77–99.
- [Thomason et al., 2017] Thomason, J., Padmakumar, A., Sinapov, J., Hart, J., Stone, P., and Mooney, R. J. (2017). Opportunistic active learning for grounding natural language descriptions. In *Conference on Robot Learning*, pages 67–76.
- [Thomason et al., 2019] Thomason, J., Padmakumar, A., Sinapov, J., Walker, N., Jiang, Y., Yedidsion, H., Hart, J., Stone, P., and Mooney, R. J. (2019). Improving grounded natural language understanding through human-robot dialog. *arXiv preprint arXiv:1903.00122*.

- [Thomason et al., 2015] Thomason, J., Zhang, S., Mooney, R. J., and Stone, P. (2015). Learning to interpret natural language commands through human-robot dialog. In *IJCAI*, pages 1923–1929.
- [Tomasello, 2010] Tomasello, M. (2010). *Origins of human communication*. MIT press.
- [Tomasello and Farrar, 1986] Tomasello, M. and Farrar, M. J. (1986). Joint attention and early language. *Child development*, pages 1454–1463.
- [Turing, 1950] Turing, A. M. (1950). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.
- [Turner, 1996] Turner, M. (1996). *The literary mind: The origins of thought and language*. Oxford University Press.
- [Varela et al., 2016] Varela, F. J., Thompson, E., and Rosch, E. (2016). *The embodied mind: Cognitive science and human experience*.
- [Vigliocco et al., 2004] Vigliocco, G., Vinson, D. P., Lewis, W., and Garrett, M. F. (2004). Representing the meanings of object and action words: The featural and unitary semantic space hypothesis. *Cognitive psychology*, 48(4):422–488.
- [Vinyals et al., 2015a] Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015a). Grammar as a foreign language. In *Advances in neural information processing systems*, pages 2773–2781.
- [Vinyals et al., 2015b] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015b). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.
- [Wang et al., 2019] Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., and Hengel, A. v. d. (2019). Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [Werbos, 1990] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.
- [Weston et al., 2015] Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.
- [Wiedemeyer, 2015] Wiedemeyer, T. (2014 – 2015). IAI Kinect2. https://github.com/code-iai/iai_kinect2. Accessed February 28, 2019.
- [Winograd, 1973] Winograd, T. (1973). A procedural model of language understanding.
- [Xiong et al., 2016] Xiong, C., Merity, S., and Socher, R. (2016). Dynamic memory networks for visual and textual question answering. In *International Conference on Machine Learning*, pages 2397–2406.
- [Xu and Saenko, 2016] Xu, H. and Saenko, K. (2016). Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer.
- [Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.
- [Yang et al., 2016] Yang, Z., He, X., Gao, J., Deng, L., and Smola, A. (2016). Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29.
- [Ye et al., 2019] Ye, L., Rochan, M., Liu, Z., and Wang, Y. (2019). Cross-modal self-attention network for referring image segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [Yi et al., 2019] Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., and Tenenbaum, J. B. (2019). Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*.
- [Yogatama et al., 2019] Yogatama, D., d’Autume, C. d. M., Connor, J., Kocisky, T., Chrzanowski, M., Kong, L., Lazaridou, A., Ling, W., Yu, L., Dyer, C., et al. (2019). Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.
- [Yu et al., 2017] Yu, H., Zhang, H., and Xu, W. (2017). A deep compositional framework for human-like language acquisition in virtual environment. *arXiv preprint arXiv:1703.09831*.
- [Yu et al., 2018a] Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., and Berg, T. L. (2018a). Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315.
- [Yu et al., 2018b] Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., and Berg, T. L. (2018b). Mattnet: Modular attention network for referring expression comprehension. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [Yu et al., 2016] Yu, L., Poirson, P., Yang, S., Berg, A. C., and Berg, T. L. (2016). Modeling context in referring expressions. *Lecture Notes in Computer Science*, page 69–85.
- [Zellers et al., 2019] Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (2019). From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6720–6731.
- [Zettlemoyer and Collins, 2007] Zettlemoyer, L. and Collins, M. (2007). Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687.

- [Zettlemoyer and Collins, 2005] Zettlemoyer, L. S. and Collins, M. (2005). Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 658–666.
- [Zhu et al., 2016] Zhu, Y., Groth, O., Bernstein, M., and Fei-Fei, L. (2016). Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004.
- [Ziemke et al., 1999] Ziemke, T., Rethinking Grounding, W., Riegler, A., Peschl, M., and von Stein, A. (1999). Understanding representation in the cognitive sciences.

Appendix A

LEARNING TO UNDERSTAND NAVIGATIONAL INSTRUCTIONS

Figure A.1 shows a map along with two paths generated by our data generator. Table A.1 lists the generated instructions for corresponding path segments. Table A.2 lists some text templates for specific task categories along with physical task patterns. Table A.3 lists the hyperparameters used in the SAIL and SAILx experiments.

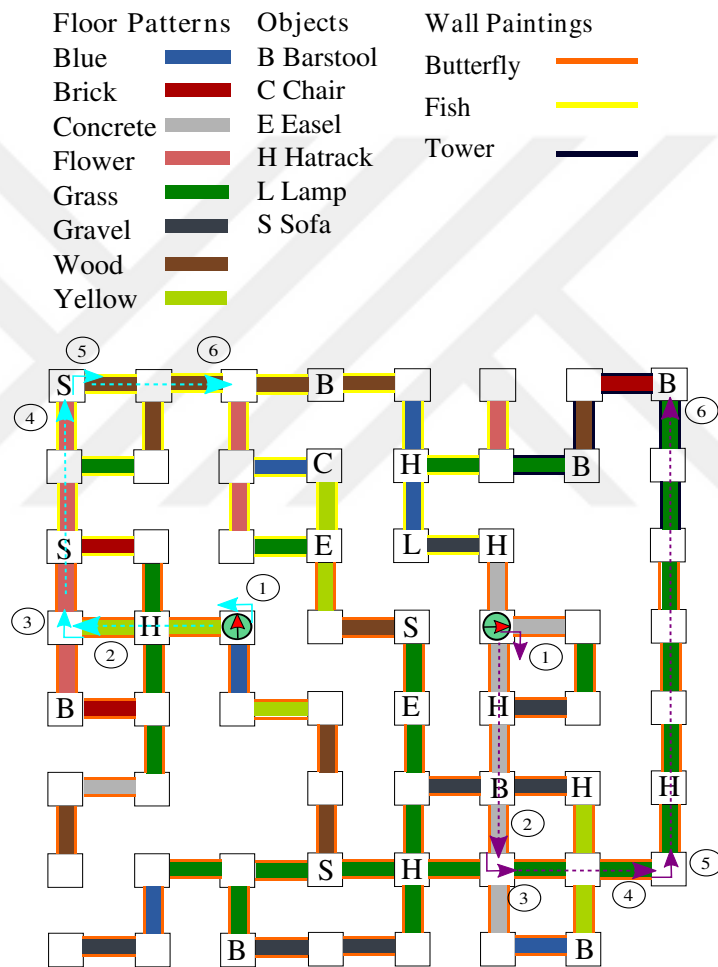


Figure A.1: An example map and two segmented paths generated by our artificial data generation framework.

Segment	Physical Task Pattern	Task	Instruction
Aqua-1	Turning to a hall contains a unique object	Turn to X	turn to the honeycomb yellow corridor containing the hat rack
Aqua-2	Passing a unique object	Move Until	move forward along the yellow corridor passing the hat rack
Aqua-3	Turning to a hall (the length is the distinguishing property)	Turn to X	turn to the long flower hall
Aqua-4	Moving to a junction (the crossed floor pattern is the distinguishing property)	Move Until	go forward until you see the brown flooring on your right
Aqua-5	x	Language Only	turn right
Aqua-6	Moving to a junction	Move Until	move until the second corridor and stop
Purple-1	Orienting at the 't' shape junction	Orient	turn so that the wall is on your right
Purple 2-3	Moving to a junction (the crossed floor pattern is the distinguishing property) + Turning to a hall (the length is the distinguishing property)	Any Combination (Move Until + Turn to X)	move along the concrete path until it crosses the green hall and look for the short alley
Purple-4	Moving to an end	Move to X	walk to the empty corner you see in front of you
Purple-5	Turning to a hall contains a unique object	Turn to X	turn to the grass containing the stool
Purple-6	Moving to a unique object	Move to X	go along the path to the stool
Stop	A unique hall around the final position	Description	to your left you should see the brown hallway

Table A.1: Example instructions generated for the segments in Figure A.1.

Task	Physical Task Pattern	Text Template
Language Only	X	turn/go/move <direction>
Language Only	X	move/go/walk forward/ahead <number>steps/blocks
Turn to X	Turning to a hall with a discriminative wall paper	"look for the"/"turn your face to the" corridor/hall/hallway with the pictures of <wall paper>
Turn to X	Turning to a unique object	"look for the"/"turn until you see the"/ "turn to the"/"turn your face to the" <object>
Move to X	Moving to a unique object	move/go/walk forward/ahead/straight "to the"/ "until you get to"/"until you reach to" <object>
Move to X	Moving to a dead end	move/go/walk forward/ahead/straight "into dead end"/"until you get to a dead end"
Move to X	Moving to other end of the corridor	move/go/walk forward/straight/ahead to the other end of the corridor/hall/path/alley
Turn and Move to X	Turning and Moving to a corner	"turn and move/go/walk"/move/go/walk "to the end of the"/ "until the end of the" corridor/hall/path
Move Until	Passing an object	move/go/walk a/one segment/block/step beyond the <object>
Move Until	Moving until the next hall	take the corridor/hall/path until the next corridor/hall/alley
Orient	Placing back to a 't' intersection	"turn so the wall is on your back"/ "place your back to the 't' intersection"
Any Combination	Moving to the end and turning	at the end of the corridor/hall/path "turn"/"take a"/"turn to the" <direction>

Table A.2: Example text templates from the data generator.

SAIL

Hyperparameters / Models	Grid	B.O.F	L.O
hidden size	120	110	100
embedding size	120	65	60
filter shapes	[(1,5), (5,5), (1,12)]	x	x
number of filters	[40, 80, 50]	x	x
dropout rate for encoder (applied to the embeddings)	0.53	0.6	0.6
dropout rates for decoder (applied to the input and output of the LSTM)	[0.5, 0.8]	[0.5 0.8]	[0.5 0.8]
coefficient for the weight initialization	13.5	0.5	0.5
hidden size of the attention network	100	x	x

SAILx

Hyperparameters / Models	Grid	B.O.F	L.O
hidden size	65	100	100
embedding size	177	60	60
filter shapes	[(1,5), (5, 12)]	x	x
number of filters	[27, 188]	x	x
dropout rate for encoder (applied to the embeddings)	0.6	0.6	0.6
dropout rates for decoder (applied to the input and output of the LSTM)	[0.5, 0.8]	[0.5 0.8]	[0.5 0.8]
coefficient for the weight initialization	10	0.5	0.5
hidden size of the attention network	36	x	x

Table A.3: Hyperparameters used in experiments.