ABSTRACT

Title of Thesis: Transfer Learning of Grounded Language Models For Use In Robotic Systems

Patrick D. Jenkins, Master of Science, 2020

Thesis directed by: Dr. Cynthia Matuszek
Department of Computer Science

Grounded language acquisition is the modeling of language as it relates to physical objects in the world. Grounded language models are useful for creating an interface between robots and humans using natural language, but are ineffective when a robot enters a novel environment due to lack of training data. I create a novel grounded language dataset by capturing multi-angle high resolution color and depth images of household objects, then collecting natural language text and speech descriptions of the objects. This dataset is used to train a model that learns associations between the descriptions and the color and depth percepts. Vision and language domains are embedded into an intermediate lower dimensional space through manifold alignment. The model consists of two simultaneously trained neural nets, one each for vision and language. Triplet loss ensures that the two spaces are closely aligned in the embedded space by attracting positive associations and repelling negative ones. First, separate models are trained using the University of Washington RGB-D and UMBC GLD datasets to get baseline results for grounded language acquisition on domestic objects. Then the baseline model trained on the

UW RGB-D data is fine tuned through a second round of training on UMBC GLD. This fine tuned model performs better than the model trained only on UMBC GLD, and in less training time. These experiments represent the first steps of the ability to transfer grounded language knowledge from previously trained models on large datasets onto new models operating on robots operating in novel domains.

APPROVAL SHEET

Title of Thesis: Transfer Learning of Grounded Language Models for Use in Robotics Systems

Name of Candidate: Patrick D. Jenkins, Master of Science, 2020

Thesis and Abstract Approved:

Dr. Cynthia Matuszek
Assistant Professor
Department of Computer Science

Date Approved:

CURRICULUM VITAE

Name:	Patrick D. Jenkins
Degree and date to be conferred:	Master of Science, 2020
Collegiate institutions attended:	University of Maryland, Baltimore County, MS in Computer Science, 2020 University of Maryland, Baltimore County, BS in Computer Science, 2015 University of Maryland, Baltimore County, BS in Mathematics, 2015
Field of study:	Computer Science
Professional positions held:	Graduate Teaching Assistant, UMBC, Sept 2019 - June 2020 Engineer Cognitive Science, Northrop Grumman Corp., Annapolis Junction, MD, July 2018 - Aug 2019 Cryptologic Computer Scientist, National Security Agency, Ft. Meade, MD, Aug 2016 - June 2018

TRANSFER LEARNING OF GROUNDED LANGUAGE MODELS FOR USE IN ROBOTIC SYSTEMS

by

Patrick D. Jenkins

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Master of Science
2020

Advisory Committee:
Dr. Cynthia Matuszek, Chair/Advisor
Dr. Francis Ferraro
Dr. Timothy Finin

Dedication

For Danya, enniku ninne ishtamanu

Acknowledgments

I would first like to thank my advisor, Dr. Cynthia Matuszek, for her support and mentorship throughout my graduate education. Her encouragement to take the leap to becoming a full time student has had more of an impact on my life than I think she knows. Thank you for guiding me through grad school and sometimes being more of a career coach than research advisor.

Thank you to IRAL's co-mentor and my defense committee member, Dr. Frank Ferraro, for your wonderful insights on language and cats, and for calling Cynthia to put money in the MTurk account from the runway. Thanks to my other committee member Dr. Tim Finin for his comforting words on Ph.D rejections and trust to fill in for his undergraduate AI class.

To all of my collaborators and colleagues in the IRAL LAB, a huge thank you. Especially Padraig Higgins who troubleshooted all of my hardware and ROS issues, Rishabh Sachdeva for handling the speech side of our dataset, Gaoussou Youssouf Kebe for sticking with me through the ups and downs of trying to move large amounts of data, Kasra Darvish for his kind words and patience as I bounced ideas around, and John Winder for advice on navigating research, grad school, applications, and academia.

Thank you to Jimmy Hawkins and Linda Obcamp for providing the support to get this whole thing started many years ago, and to Dr. Randall Dahlberg who took a chance on a young intern and gave me my first taste of research.

Thank you to my family and friends who didn't always understand what I was

doing but cheered me on anyway.

Finally, an unfathomable amount of thanks to my partner Danya Murali. I really couldn't have done this without you. You celebrated my highs and pulled me through my lows. You allowed me to take a risk on myself that I will forever be grateful for.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API	Application Programming Interface
AMT	Amazon Mechanical Turk
BLEU	Bilingual Evaluation Understudy
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CV	Computer Vision
GLD	Grounded Language Dataset
HIT	Human Intelligence Task
HRI	Human Robot Interaction
IR	Infrared
IRAL	Interactive Robotics and Language Lab
LSTM	Long Short-Term Memory
MRM	Mean Reciprocal Rank
MTMs	Mechanical Turk Masters
NIR	Near Infrared
NLP	Natural Language Processing
POS	Part of Speech
PCL	Point Cloud Library
ROS	Robot Operating System
ToF	Time-of-Flight

Chapter 1: Introduction

The allure of having an in home robot assisting with tasks such as cleaning, cooking, and caretaking is obvious. The time and effort saved by automating chores with an entity that does not tire or complain would improve many lives [1–3]. You could get home from work where your robotic chef has already prepared a meal for you and passed off the dishes to another robot, freeing you to watch TV or play with the kids. Meanwhile, your grandmother who can't get up and down the stairs as well as she used to was able to retrieve her medication from a robot who scurries around the house. Another robot has finished cleaning and sorting the toys the kids left out from last weekend's play date.

One of the challenges of creating such robots is that peoples' houses often have unique objects that the robot must be able to interact with. For example, a cooking robot would need to use the many tools within a users' kitchen to prepare meals. But how often have you entered someone's kitchen and found a gadget you weren't familiar with Or a gadget that you were familiar with in a completely different form? If a robot is presented with the same challenge, it will need to either learn about the object, through asking questions, or already have the ability to understand the object based on other things it knows about. Complicating this scenario is the fact

that kitchens around the world look very different, from the forks, knives, and stand mixers of Western cultures to the woks, chopsticks, and chapati presses of Eastern cultures. And this is just one environment that a robot might operate in. A whole new set of objects and challenges are encountered in office and living spaces.

Grounded language acquisition can alleviate some, but not all, of these challenges. Grounded language acquisition is the process of joining natural language with sensory input from the physical world [4–8]. By *grounding* the language on to physical objects and senses, the robot attains a deeper understanding of its environment by not only being able to identify the objects, but relate their attributes to other objects and senses. For example, a chef's knife might be described as "sharp, long, and used to cut things". But when presented with a bread knife, prior language knowledge not only allows the robot to identify it as a knife, it also understands the edge is serrated, or that it has a different use than the chef's knife. This can give the robot a jump start on interacting with novel objects, but still requires that it be taught about these objects.

One approach might be to have each person annotate the things in their environment for the robot. The main problem is that in order for the robot to become accurate enough through requires many more training examples than a typical user would want to annotate and would need to be performed every time a new object is added to the home [9]. Even within the small dataset presented in this work, accuracy was only achieved after thousands of training examples which would be infeasible for a human. One possible solution is for robots to teach other robots what they know [4] either through transfer learning or domain adaptation. Transfer

learning and domain adaptation are two machine learning techniques that use prior training that is then adapted to related tasks or datasets. This not only reduces the volume of training data needed for the end task, but can outperform models trained solely on the end task due to greater context of related tasks.

Continuing with the kitchen example, imagine a robot trained on a large database of common household items. For most purposes this robot could operate in a kitchen, but might fail if the kitchen was inside a bakery due to specialized tools. With a smaller dataset of bakery items, the robot could fine tune its kitchen model to adapt to its environment [10]. The robot would take what it already knows about bowls, utensils, ingredients and modify its knowledge for the large industrial mixing bowls, bench scrapers, and flour of the bakery. Furthermore, this dataset could be generated from other bakery robots, eliminating the need for human annotation.

In this thesis I present UMBC GLD, a novel grounded language dataset of RGB and depth images of common household objects and descriptions of those objects in natural language text and speech. This dataset represents the smaller fine tuning dataset from the bakery example. To simulate the larger datset, I use a similar RGB-D object dataset from the University of Washington [11] with paired text descriptions. Manifold alignment [12] with triplet loss [13] is used to train grounded language models on both datasets. Finally, I transfer the grounded language knowledge learned from the UW dataset onto a model that understands the objects within UMBC GLD. While manifold alignment and transfer learning are known techniques, to my knowledge this represents the first time that they have been used together

with grounded language acquisition.

Chapter 2: Related Work

In this chapter I describe transfer learning and domain adaptation as well as discuss some of the previous work in the Computer Vision (CV) and Natural Language Processing (NLP) fields and how they relate to current work in grounded language acquisition.

2.1 Computer Vision

The field of Computer Vision (CV) is concerned with the techniques and processes through which a computer "sees", or processes visual percepts. These raw percepts range from the color spectrum that many humans are familiar with, to infrared (IR) and various forms of depth sensing [14]. In this section I describe two tasks that are important for grounded language, object classification and scene segmentation.

2.1.1 Object Classification

Object classification is easy to describe yet difficult to achieve. Put simply, it is the task of labeling the primary object within an image. The labels here refer to human annotations, usually nouns, of the contents of the image.

A popular algorithm for solving this task is convolutional neural networks (CNNs) [15–17]. CNNs are neural networks that have specialized hidden layers. Instead of the typical neural network which takes input from the entire image, convolutions use kernels to capture local information within the pictures such as edges, textures, and patterns [15, 16]. In this work, CNNs are used to extract visual features since the hidden layers can be thought of as complex feature extractors [18].

Possibly the most famous object classification challenge or benchmark is ImageNet [19, 20]. ImageNet now contains about 100,000 "synsets" (roughly a label, but images can be part of multiple synsets) with around 1000 images representing each synset. Interestingly, CNNs are still a very popular algorithm for this work with Google reaching state of the art with their complex GoogLeNet architecture [17]. Eitel et al used ImageNet to pretrain their RGB and depth multimodal feature extractor that is used in this thesis [18].

There is a subtle difference between object classification and grounded language acquisition. Object classification can only tell us what label is associated with the image at hand, and that label must come from the possible values within the training data. Meanwhile, grounded language acquisition tells us how close language is to describing a certain object through language modeling. The language and the object need not be part of the training data since the description is passed through a language model. Alternatively, some work uses GL to generate language from visual percepts [6, 21].

2.1.2 Scene Segmentation

The real world is messy and cluttered. In order to operate in the real world, robots need to be able to make sense of the mess by using segmenting to break it into pieces they can understand [22, 23]. Once the scene is broken up, the robot can use the visual inputs from each segment to run object classification or a grounded language model to label or otherwise process the visual percepts. Scene segmentation is also an important part of navigation in robots since it can be used to find navigable surfaces such as roads [24] or break up indoor scenes to find walls, floors, and obstacles [25].

Silberman and Fergus [25] explored scene segmentation of indoor spaces using a Kinect depth sensor. This is promising for robotics since many research platforms use a Kinect for visual perception. Within this thesis I used a Kinect to capture depth information of objects, which means the same platform can be used for both grounded language acquisition and scene segmentation.

This thesis does not make any direct use of scene segmentation since all of the data was collected as individual objects. However, it is plausible that in future work a platform could segment a scene and feed inputs into a grounded language model to obtain a richer understanding of the world around it.

2.2 Natural Language Processing

Natural language is incredibly complex and the field of Natural Language Processing is aptly broad and often just as deep. While there are two main domains

that NLP is concerned with, text and speech, the number of applications and techniques are many [26–30]. In this section I will mainly discuss tasks that are relevant to grounded language. If the end goal is to communicate effectively with a robot, then the robot will need to understand the natural language inputs that we give it [31, 32] . Depending on the robot, it may also need to generate its own language when asking for help, clarifying, or communicating internal states [21, 33].

2.2.1 Part of Speech Tagging

Part of Speech (POS) tagging is the task of automatically annotating words or tokens of a text with their corresponding parts of speech such as Noun, Verb, Adjective, etc. Often these high level categories are broken down into more specific classifications such as Proper Noun or Past Tense Verb [34]. This task is complicated (at least in English) by homonyms and proper nouns which could have multiple correct tags depending on the context. POS tagging is commonly used as a preprocessing step or subtask of a larger NLP task. This is because parts of speech are often useful features for other down stream tasks such as language modeling, phrase recognition, and [4, 35].

The techniques for POS tagging are numerous. Brill created a simple rules based POS tagger that does not use any previous knowledge of the syntax to perform as well as stochastic methods of the time [36]. Shortly after, Schmid developed a neural network solution which takes a word and surrounding words as input to the net and outputs a classification vector where the greatest activation relates to a

part of speech [37]. Brants' used a second order Markov model which further uses the context of the sentence as a whole by accounting for probabilities of transitions between parts of speech [38].

Today, the state-of-the-art algorithm for POS tagging is Akbik et al.'s [39] highly complex Flair tagger[1]. Flair uses a Bidirectional, Long Short-Term Memory (LSTM) with an additional conditional random field (CRF) decoding layer [40]. Bidirectional means that the model can look at the features of past and future parts of the sequence, in this case words. An LSTM is a recurrent neural network model that can store past inputs to influence how a sequence is classified or generated [41, 42]. Finally the CRF layer segments and labels the outputs of the LSTM [43].

2.2.2 Sentiment analysis

The more abstract field of sentiment analysis is concerned with interpreting the mood, emotions, or attitudes of a speaker through language [44]. This NLP task is particularly important in the field of Human Robot Interaction (HRI) where a robot may want to adapt its behavior based on how the human is interacting with it [45–47].

One of the benchmark datasets of sentiment analysis is the Stanford Sentiment Treebank (SST-5) which contains sentences rated on a 5-point scale of "very negative" to "very positive" [48]. State-of-the-art NLP algorithms [49, 50] achieve roughly 50-55% accuracy on when classifying the testing set of SST-5. One feature of language that makes sentiment analysis so difficult is sarcasm [51]. However,

[1] https://github.com/flairNLP/flair

being able to detect and understand sarcasm via text is becoming more important, especially for understanding sentiment and meaning of online forums such as Twitter and Facebook. These platforms host "bots" that attempt to influence people and identifying and understanding them is crucial to understanding how people communicate online [52, 53].

2.2.3 Language Modeling

Language modeling is simply computing probabilities of sequences of words or sentences. This is a crucial task for robots interacting with humans both when generating and understanding their language. Tellex et al used inverse semantics to find utterances for robots to ask for help while building furniture [21]. Roy maximized the chances that a sentence would describe a target rectangle [8]. Many language models use some form of LSTM but graphical models can also be effective [54–56].

In this work I use BERT as a language model to extract feature vectors from text descriptions of object [55]. As discussed later in 5.2.1, BERT is a pretrained English language model that works on multiple sentence documents to produce high dimensional word embeddings.

2.2.4 Question Answering

Answering questions is natural for humans, but the task of parsing through a question, understanding what it is asking, and generating the language in the form of a response is a whole field of NLP. In robotics, the inquirer is often the human

who needs help with a task. However, Tellex et al. [21] used grounded language to generate requests for help in building furniture from human partners and Pillai et al. [9] used question answering to improve annotation tasks.

This work does not make use of any question answering systems but is important to metion since question answering is a potential solution to annotating novel objects for robotic systems [9]. Additionally, the dataset that I create addresses some, but not all, of the issues of question answering datasets such as CLEVR [57], Scene and GeoQA [58] discussed in Section 2.5.3.

2.2.5 Automated Machine Translation

In an increasingly global world it is imperative that domestic robots are able to be deployed in any country. In the context of communicating with these robots using natural language, the apparent roadblock is modeling and translating multiple languages. Most people are probably familiar with machine translation in the form of Google Translate [59][2]. Machine translation is the task of automating translation of one language into another [60]. And while this work does not make any direct use of machine translation, it is worth discussing in the context of grounded language and domestic robots.

Beyond the task of modeling individual languages, language can affect how people ground objects within their mind [61]. The "Bouba/Kiki effect" [62, 63] suggests that humans connect attributes and characteristics to sound and language. But even within a seemingly global similarity, there can be cultural differences [64].

[2] https://translate.google.com/

Far too often, the scientific community focuses on English corpera for training NLP and grounded language models. This is both because English is the lingua franca of scientific research and it can be easier to use one of the many readily available English corpera rather than create a new dataset in a different language. Despite this, some work has been done to move multilingual grounded language forward.

Chen et al. [65] expanded upon their previous RoboCup sportscasting [66] to cast games in both English and Korean without any prior language specific knowledge. This work showed that grounded language acquisition could be done between multiple languages. Kery took this further by exploring what model or preprocessing changes are critical to consider when building such systems [67, 68].

2.3 Grounded Language Acquisition

Grounded Language (GL) is a field of computer science that combines aspects from CV and NLP to learn an association between the language and visual perception domains [4–7]. Learning these associations between the object and the label instead of just the label for an entity is what makes GL unique amongst these fields. In a sense, an effective grounded language model could describe a novel object by understanding the links between the visual inputs it receives and the way our language is structured when those visual inputs are present. Alternatively, it could take language as input and through those associations select or find an example of that language within the world around it. The link between language and the physical world is what makes grounded language acquisition a promising solution

for communicating effectively with robots, systems that are by definition interfacing with the physical world.

One of the first experiments in grounded language acquisition was to train a system to select colored rectangles from an arbitrary scene from descriptions of a target rectangle [8]. The language sometimes described the target using the size, color, or its relationship in space to other rectangles. Through a combination of a language model and visual feature extraction, the system was able to do more than label the colors of each rectangle in the scene, it was able to understand the meaning of the language as it related to the scene. Mooney used this idea to train a RoboCup (a robotic version of soccer) sportscasting agent [6]. The agent could perceive a game states of RoboCup, and output sentences that would describe action within the game.

The idea of an agent understanding action, tasks, and sequences is foundational to how we would like to interact with robots. In order for them to clean, cook, shop, and take care of us [1, 69], they need to be able to understand the directions and tasks we give them without our demonstrating them first [70].

2.4 Transfer Learning

Transfer learning is a machine learning technique for training a neural network with less training data [10, 71]. The reason this is helpful is because for many tasks there is not a lot of data. As discussed in Section 2.2.5, models in other languages are useful but there is often not enough data to sufficiently train. Transfer learning

helps get around this problem by first training the neural network on a large set of data that is accessible and then performing a fine tuning step by training on data from the target domain.

A common example is a neural network that is trained to recognize road signs [72, 73]. Road signs are common in many countries, yet because of idiosyncrasies in infrastructure and driving laws are not globally uniform. For example, in Germany a priority road is indicated by a yellow diamond inside a white border. A network trained on US road signs would misclassify this sign. However, that doesn't mean that this network hasn't learned useful information about classifying road sings such as edge detection and color pattern recognition. This knowledge can be leveraged as a starting point of training on a different dataset such as German road signs. The net will retain some of its feature finding qualities without having to relearn them and finetune its weights to perform classification on the new dataset. The advantages to this technique are that the finetuning dataset need not be as a large as the primary dataset, and the net doesn't need to be trained for quite as long since the weights are being tuned instead of learned.

In Chapter 5.4 I discuss a transfer learning experiment with a grounded language acquisition model. The goal of the experiment is to show that transfer learning is a viable technique for improving accuracy and training times of grounded language models.

2.5 Grounded Language Datasets

In this section I describe various useful datasets for grounded language learning. These datasets cover many tasks including object classification [11], question answering [57, 58], direction following [74, 75], and instruction following [56]. Their tasks and collection methods uncover many challenges of creating grounded language datasets including the expense (both in time and resources), the need to be augmented by generated data, or the limited scopes in which the data can be used.

2.5.1 UW RGB-D Dataset

The UW RGB-D dataset [11] is a large-scale data set built for training object recognition models. The dataset contains over 250,000 RGB-D images of 300 unique objects from 51 categories. In addition to the color and depth images, the authors also released segmentation masks for each image. This allows for the removal of background noise from the images in order to improve accuracy of classification.

While initially intended for an object classification task, there are many computer vision datasets that can be augmented with language in order to enable grounded language learning. Recently, the UW-RGBD data has been used in conjunction with descriptions of the objects collected from Amazon Mechanical Turk to explore category free grounded language [76]. However, this requires other researchers to collect their own natural language descriptions, sometimes with annotations, which can be expensive and time consuming [77].

I use this dataset with previously collected language descriptions as a basis for

transfer learning to a dataset I collected that is discussed in Chapter 3.

2.5.2 Rectangle Descriptions

One of the earliest experiments in grounded language acquisition was Roy's rectangle description task [8]. The goal was to learn the semantics of language that describes a computer generated scene. Each scene consisted of 10 randomly sized and colored rectangles. Visual features, including RGB values, position, and size, of each rectangle within the scene were "extracted." The paired language describes a target rectangle within the scene. Sometimes the paired language was simple and contained few referents, while other times it was complex, with multiple referents, spatial descriptions, and adjectives.

The visual data for this task can easily be generated by a computer program. However, the language data, as in most tasks, is much more expensive to collect since it requires a human in the loop and sometimes post-processing to transform the raw data into something usable by researchers. For this particular task an undergraduate recorded, at a very fast pace, 518 utterances over three hours. The audio from this collection was then manually transcribed into text. The manual audio transcription task can take anywhere between four and ten hours per hour of audio depending on the quality of the audio being transcribed and the final quality of the transcription [78–80].

2.5.3 CLEVR, Scene, and GeoQA

The CLEVR dataset [57] consists of scenes of objects with annotations for color, shape, size, and spatial relation to other objects within the scene. It was designed to test a range of question answering and machine reasoning tasks. However, it is not very useful when generalizing to real world scenes. The objects within the dataset are toy blocks that are monochromatic with well defined shapes such as pyramids, cubes, and spheres. This makes the choices for the annotation task very limited which in turn limits the types of questions and reasoning that could be done on the scenes. For the limits of grounded language systems to be properly exercised, there is a need for the datasets to more closely approximate the real world with more realistic cluttered scenes.

Krishnamurthy [58] addressed this in their work on a question answering system with their Scene and GeoQA datasets. Scene contains pictures of real world office supplies arranged on a desk in various arrangements. Questions can then be answered about the location of objects relative to others within the scene. GeoQA is intended to test the same tasks with well annotated maps.

2.5.4 SAIL and Direction Following

Following directions through an environment is a necessary component of robotic systems [81]. The SAIL dataset [74, 82] was used as part of an early experiment into following direction using grounded language. The spaces were created as virtual environments of hallways with identifying pictures on the walls. Par-

ticipants were given time to learn these environments and then provide written directions to get to landmarks within the environment. The virtual environments and the 682 route instructions make up the corpus. Despite the environments not being representative of the real world and flaws within the route instructions pointed out in [74], this dataset is still in use today [83].

Matuszek, Fox, and Koscher [75] created a platform that uses a laser range-finder to scan its environment. From the scan it uses Voronoi Random Fields [84] to automatically label the map with an accuracy up to 90%. This eliminates the need to generate and annotate maps and allows robots to work directly in their environments. To compliment the maps, paths were generated and volunteers were asked to provide directions for the robot to take the path. Eight volunteers were asked to describe five paths, yielding a small natural language set of directions. However, the authors augmented this small set by synthesizing short directional phrases from the natural language. Thus the authors were able to create a robot that understands natural language directions without the need of humans to create or annotate a large training set.

2.5.5 Forklift Actions and Commands

Tellex et al [56] collected data about asking a virtual robotic forklift to perform various tasks. They did so by asking Turkers to watch a video of the forklift performing an action. After watching the video they would type in a command that could be given to a forklift operator to perform the demonstrated action. While

very clever, this approach is narrow in its scope. It can only be applied to what was included in the virtual environment including forklifts and the handful of objects the forklift could interact with.

Recently, Shridhar et al. [70] have attempted to correct some of these short comings by releasing ALFRED, a benchmark dataset for understanding commands for every day tasks.

Chapter 3: UMBC Grounded Language Dataset

In this chapter I describe a novel Grounded Language Dataset (GLD). The intent is to use this dataset to train a grounded language model for a domestic robot. The robot would then be able to perform household tasks and chores such as retrieving first aid materials and medicine, repairing broken items, building furniture, organizing, cleaning, and cooking [3].

The dataset is made up of high resolutions images and point clouds (color + depth) of common household objects from five high level categories of *food*, *home*, *medical*, *office*, and *tool*. These categories closely match the similar UW RGB-D dataset, but differ with the additional focus on medical supplies and smaller items due to the higher fidelity of the Kinect 3 [11]. The images and point clouds are captured from multiple angles since there is no guarantee that a robot would have the same point of view as a human in the same space. Additionally, the visual data is paired with natural language descriptions from the text and speech domains. This pairing of visual percepts and natural language constitutes a novel set of data that supports grounded language research.

This dataset address some of the challenges presented in Chapter 2.5 in a number of ways. First, it attempts to capture varied sets of domestic objects through

Topic	Classes of Objects
food	*potato, soda bottle, water bottle, apple, banana, bell pepper, food can, food jar, lemon, lime, onion*
home	*book, can opener, eye glasses, fork, shampoo, sponge, spoon, toothbrush, toothpaste, bowl, cap, cell phone, coffee mug, hand towel, tissue box, plate*
medical	*band aid, gauze, medicine bottle, pill cutter, prescription medicine bottle, syringe*
office	*mouse, pencil, picture frame, scissors, stapler, marker, notebook*
tool	*allen wrench, hammer, measuring tape, pliers, screwdriver, lightbulb*

Table 3.1: The classes of objects that appear in GLD, grouped by their high-level category.

its high level categories. This widens the scope of use to in home care such as cooking and caretaking via the food and medicine categories, but also to domestic work environments through the tools and office categories. Second, I provide the natural language descriptions both in text and speech. This saves other researchers time and cost of collecting and transcribing natural language which can be a burden as discussed in Section 2.5.2. Finally, the intent is for this data to be used for a number of tasks. The language corpus only contains descriptions of the objects which can be used to ask and answer questions, similar to the CLEVR dataset [57], or can be used to target commands involving the objects similar to Tellex et al's work [56].

This work was completed jointly with Padraig Higgins, Rishabh Sachdeva, and John Winder of the IRAL Lab. Higgins provided hardware support for the Kinect 3

and ROS. Sachdeva was responsible for collecting and analyzing the collected speech descriptions. Winder helped with high level support of data collection and direction of the work.

3.1 Image Collection

The visual data were collected using a Microsoft Azure Kinect, colloquially known as a Kinect 3, using Microsoft's Azure Kinect drivers for the Robot Operating System (ROS) [85].[1] The Kinect 3 is an RGB-D camera consisting of both a color camera and a Time-of-Flight (ToF) depth camera which enables it to capture high-fidelity point cloud data. The Kinect Azure depth camera uses the Amplitude Modulated Continuous Wave (AMCW) Time-of-Flight (ToF) principle . Near infrared (NIR) emitters on the camera illuminate the scene with modulated NIR light and the camera calculates the time of flight for the light to return to the camera. From this a depth image can be built converting the time of light to distance and then encoded into a monochromatic depth image. ROS allows for the registration of the color and depth images, matching pixels in the color to pixels in the depth image, to build a colored point cloud of the scene.

The camera was placed 60 cm away from, and 30 cm above the turn able at an angle of 30 degrees as seen in Figure 3.1. I collected a video for approximately four instances per each of the 47 classes for at total of 207 instances. The camera records one revolution of the turntable or 90 seconds at 5 frames per second for each object, yielding 450 frames from different angles. Without any post processing the

[1] https://docs.microsoft.com/en-us/azure/kinect-dk/

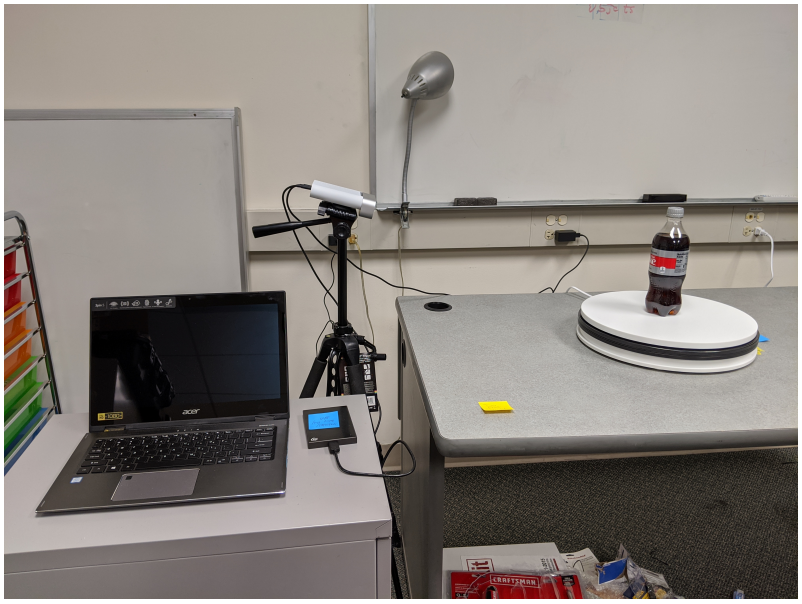


Figure 3.1: Set up for the visual object data collection. Objects are placed on the motorized turntable and imaged by the Kinect 3 for one revolution or 90 seconds. At a frame rate of 5 Hz, this yields 450 RGB, depth, and point cloud images of each object from different angles.

raw data contains information about the background which is not related to the dataset that is being collected. Point Cloud Library (PCL) [86] passthrough filters were used to crop the raw point cloud to only include the object being collected and the turntable. The point cloud is decomposed further into .png files of the color and depth channels. Four frames were selected for each instance to be included in the description labeling task. The final visual corpus thus contains 825 instance frames with the associated raw image .png, color channel .png, depth channel .png, and the original point cloud.

The depth channel is necessary in order to ground descriptions of shapes of objects and improve the accuracy of object recognition [11, 87, 88]. While I use a multi-modal approach to extracting features from the visual percepts based on Richards', Matuszek's, Eitel's, et al's work [18, 76], it is a common task to identify

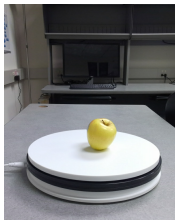
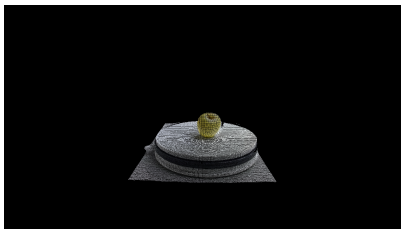
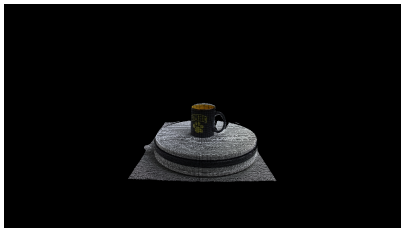
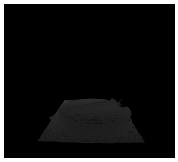
Instance	Raw Image	Point Cloud (RGB + Depth)	Depth
apple_1_1			
coffee_mug_2_3			
hammer_6_2			

Table 3.2: Examples from the visual data collection. From top to bottom: apple_1_1, coffee_mug_2_3, and hammer_6_2. The first number indicates the instance number, the second number indicates the selected frame. Left to Right: instance name, raw image, point cloud, and colorized depth.

Figure 3.2: An example of text and transcribed speech associated with a *mug* instance. The speech descriptions have some noise in them due to imperfections in Google's Speech-to-Text algorithm.

the shapes and poses of objects through their depth features [89].

3.2 Description Collection

Both the text and speech descriptions were gathered using crowd sourcing on Amazon Mechanical Turk (AMT). Mechanical Turk allows users to publish annotation tasks for users all over the world to complete. While some have criticized collecting data this way as unreliable or not as representative as collecting data in person, this is generally found to be inaccurate [90, 91]. Mechanical Turk protects each Turker's privacy by anonymizing each worker with a Worker ID. Although it has been shown that there are techniques that can uncover a worker's identity through this Worker ID [92], the main concern is unethical gathering of personal information through the Mechanical Turk task itself. Since my task does not collect private information and the Worker IDs are not published, the risk to privacy for Turkers working on my task is low.

3.2.1 Text Descriptions

Each AMT text Human Intelligence Task (HIT) includes five images of random instance frames from the visual corpus. Workers were then asked to type a description of each frame as if they were describing it to another person in one to two short sentences without describing the turntable itself or the background. An example of the HIT can be seen in Figure 3.3. Each task was assigned 10 times for a final text corpus of 40 descriptions per instance and over 8000 total text descriptions.

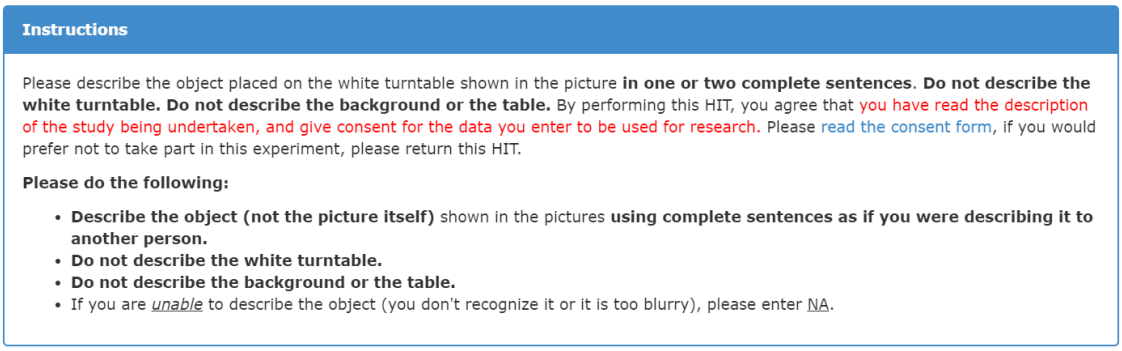


Figure 3.3: An example of the text description HIT. Turkers were asked to describe each object in the HIT.

In order to work on my task Turkers must be in the United States, have a HIT Approval Rate of 98% or greater, and have had at least 50 HITS approved. These constraints were put in place to have a higher chance of getting high quality Turkers who were native English speakers. Workers were paid $0.13 per HIT at a conservative estimated minute per HIT to pay above US Federal minimum wage. This is in line with recommendations of compensation from Lovett, et al [93] based on their survey of Mechanical Turk Masters (MTMs). While I did not use MTMs for this task, it is still important to compensate Turkers fairly.

3.2.2 Speech Descriptions

Humans use speech as a natural interface for communication. With improvements in speech to text technologies, humans will increase the use of speech to communicate with technology such as robots and smart speakers. Therefore, I collected speech descriptions of the objects as well. This data is not only useful for training grounded language models, but is useful for comparing how humans describe things within different domains.

A user interface was developed to collect spoken natural language data using MediaStream recording API.[2] A similar approach is reported in recent work [94,95] to collect data using web-based and mobile application-based systems. The interface was embedded into Amazon Mechanical Turk, and the recorded audio files were collected from these tasks. The FFmpeg library[3] was used to add the missing metadata from the audio files to make them compatible with Google's Speech-to-Text API[4] which was used to transcribe the audio. As discussed in Section 2.5.2, manual transcription of audio data can take large amounts of time [78–80]. Since there is a heavy reliance on Google's API, analysis was done to evaluate its effectiveness and is discussed in Section 4.1.

Each Mechanical Turk task included one image and five assignments were assigned for each task, an example can be seen in Figure **??**. 4059 audio descriptions were collected in total. The final dataset includes both the original speech files in .wav format as well as the text transcriptions from Google Speech-to-Text.

[2] https://developer.mozilla.org/en-US/docs/Web/API/MediaStream_Recording_API
[3] https://www.ffmpeg.org/
[4] https://cloud.google.com/speech-to-text

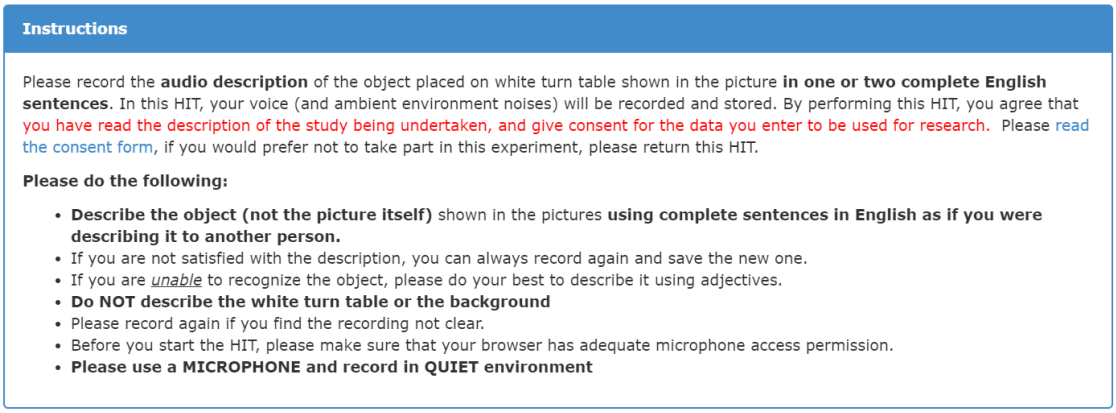


Figure 3.4: An example of the speech description HIT. Turkers were asked to record a description of each image.

Chapter 4: Data Analysis

4.1 Quality of Descriptions

One of the challenges of crowd sourcing data is being sure of the quality of the collected data [93]. To test the quality of UMBC GLD's text descriptions I randomly selected 800, around 10%, and qualitatively annotated them on a four point scale, 4 being the highest quality and 1 being the lowest. A more detailed description of the scale with examples of each class is given below.

4: Good and detailed (Accurate and detailed)

- toothbrush_2_3: It is a toothbrush with a white and pink handle with blue bristles.
- lime_3_4: This is a small green lime. It has a sticker on it with a barcode.
- measuring_tape_3_4: This is a pink and black retractable tape measure

3: Ok descriptions (Accurate but not detailed)

- plate_3_2: It's a round object that you eat on.
- stapler_1_3: This is a pink stapler

2: Bad or inaccurate (Good faith attempt that may be inaccurate or non-descriptive)

- lime_2_1: this is a lime.

1: Unusable or technical error (Critical spelling errors, Turker filled in the wrong text box, described the wrong object, answered in a language other than English, etc.)

- plate_2_1: SERVE A FOOD.

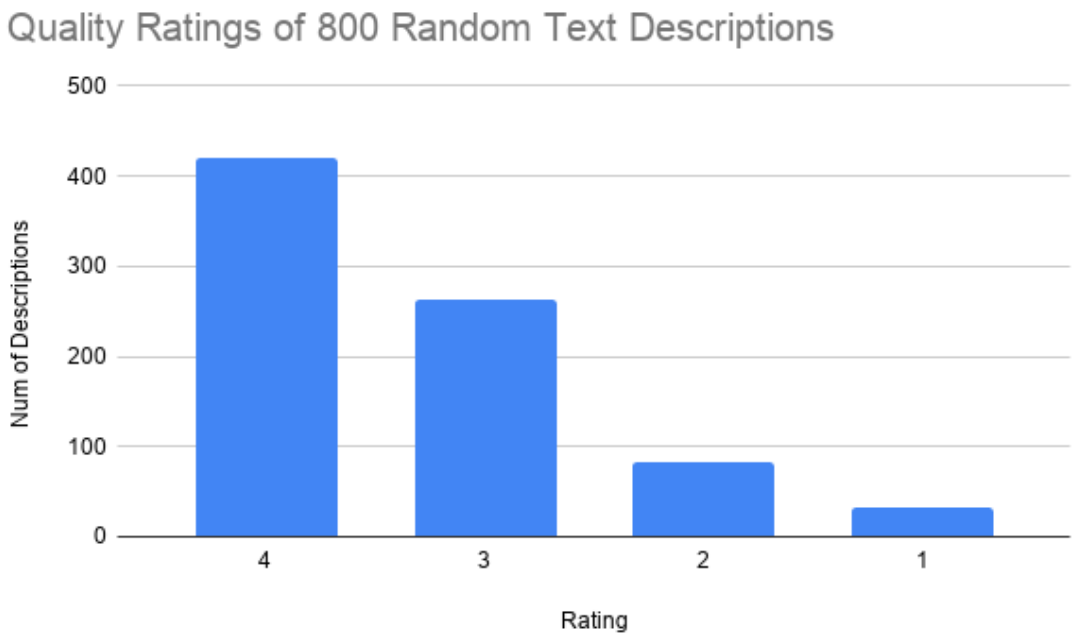


Figure 4.1: A histogram of the text description quality scores of 800 random samples.

- flashlight_3_4: a laser measuring tape

Of the 800 annotated descriptions the mean score was 3.34 out of 4, with a standard deviation of 0.82. This indicates that many of the annotators provided accurate and detailed descriptions, useful for performing grounded language acquisition. Additionally, 496 of the descriptions used complete sentences. Many of the descriptions marked as incomplete sentences were missing a verb but were still descriptive such as "A small black book with an elastic band closure." These types of descriptions are still useful for learning grounded language acquisition since they provide many descriptive phrases of the object.

A flaw in the analysis of the text descriptions was that I performed the annotations myself and was also fully aware of the end usage. This sometimes clouded my judgement when determining annotations of descriptions to rank them higher or

lower. Therefore, while showing promise, this analysis will need to be redone with an impartial judge who is not as familiar with the task, either in person or through AMT.

4.1.1 Speech Quality

The speech collection and analysis was done by my collaborator Sachdeva [96] and is a large part of his master's thesis. I have included the top level analysis for completeness but more information can be found in his thesis.

The audio file transcriptions were similarly quality checked with a four point scale. These ratings are an indication of the quality of the transcription, not the accuracy of the description of the object. It is important to note that since this scale is different from the text description annotation task that the numbers reported here cannot be compared to one another to say that text or speech was of higher quality than the other.

4: Perfect transcription (accurate transcription and no errors)

3: Pretty good transcription (main object correctly defined)

2: Slightly wrong transcription (missing keywords/concepts)

1: Wrong transcription or gibberish/unusable sound file

Of the 100 audio files that were checked, the mean rating of a sample was 3.277 with 23% of samples considered poor quality, falling within categories 1 or 2. Interestingly, the quality dropped significantly when the number of words in the transcription was less than three. This could be a byproduct of the speech-to-text algorithm failing to find natural word breaks and transcribing longer words.

The accuracy of the transcription was checked separately using a bilingual evaluation understudy, or BLEU, score [97]. This score measures how well a sentence matches a reference sentence by computing n-gram precision. BLEU was originally intended for machine translation tasks and so this is an imperfect measure for transcription. However, n-gram precision can still give a sense of how close two transcriptions are to one another. For the 100 checked transcriptions, a human annotator transcribed each audio file to be used as the reference. The average BLEU score for all transcriptions was 0.798. A score of 1 would be word for word matching and so there is an indication that the speech-to-text software is working fairly well at transcribing.

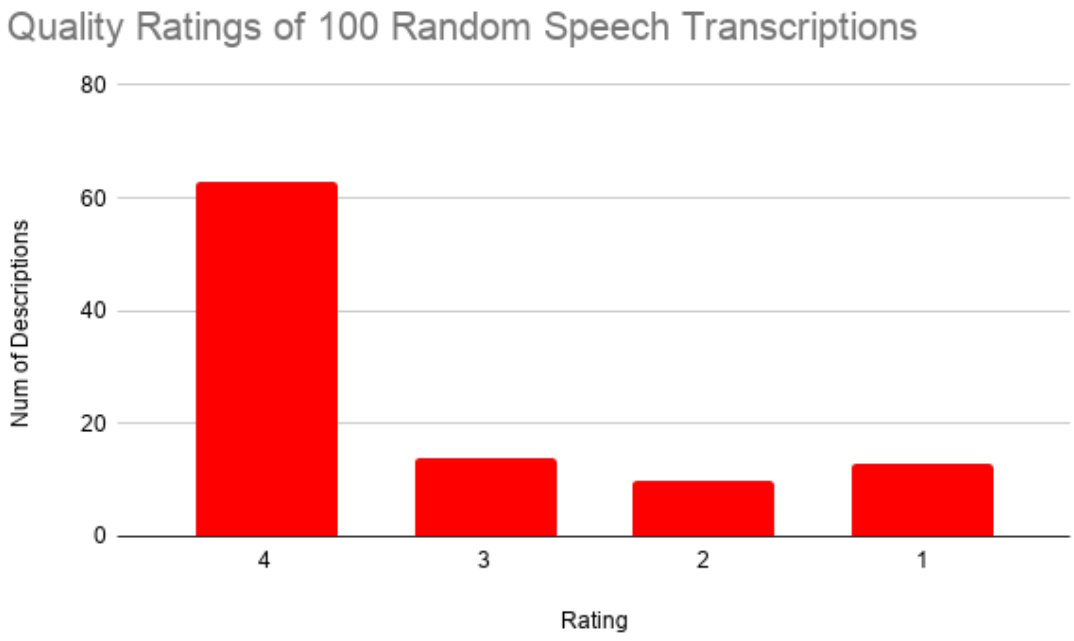


Figure 4.2: A histogram of the sound description quality scores of 100 random samples. The mean score is 3.277 which indicates decent quality. 23% of the samples are considered poor quality.

4.2 Language Analysis

In order to uncover any differences in the text and speech descriptions that might inform training techniques for grounded language models, I analyze the collected text and speech descriptions for any important characteristics. Specifically, I analyzed both the descriptions for the number of words used as well as mentions of color, shape and object name. Color, shape, and object names were important to track since they have been used as features to train grounded language models [98]. Even as grounded language acquisition moves into category-free learning, it is interesting to examine how often color, shape, and object names are used in descriptions of images [76, 99].

I gathered a list of 30 common color terms from large language corpora and compared each description to see if it included one of the common colors [100]. Similarly, I use a vocabulary list of shape terms to count how many descriptions included shape descriptions. It is worth noting that shape descriptions are less well defined than colors and that a better vocabulary of shape descriptions would be helpful towards this kind of analysis. Finally, I count the number of times the object name is used in a description.

It was initially hypothesized that people would use more words when describing objects through speech than text because it is lower effort to talk than to type. As a side effect of this hypothesis, the frequency of color, shape, and object names would be higher in speech than in text. However, there is no significant differences in the average length of descriptions between speech and text. In fact, while speech has

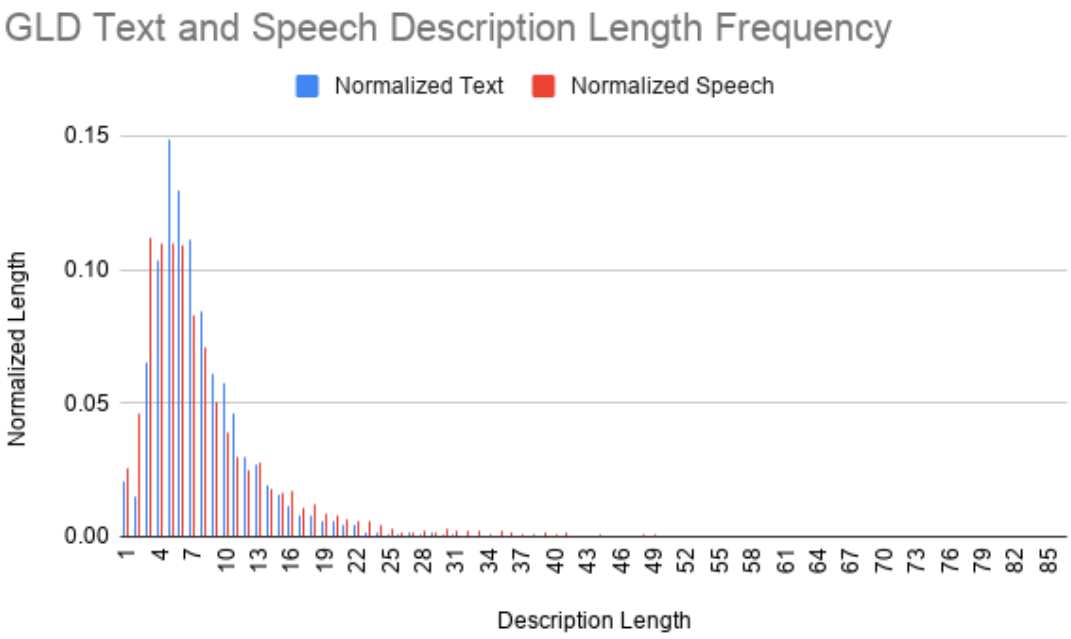


Figure 4.3: Sentence length (number of words) frequencies of text and speech descriptions normalized by the total number of text or speech descriptions. The averages for each were about the same at $\mu_{text} = 7.8, \mu_{speech} = 8.7$.

slightly more average words per description at 8.7 compared to text at 7.8, when stop words are removed the averages are 4.7 and 5.0 respectively. The larger drop in number of words from the speech descriptions may be due to filler words captured with the speech-to-text API. There was also no significant difference in the frequency distributions of mentions of color, shape, or object name between the two modes.

Table 4.1 shows the most frequent tokens in text and spoken data. Most of the tokens are consistent in both cases, with color appearing as the most common choice to describe the objects. The difference in magnitude of counts is because there are almost twice as many textual descriptions as speech descriptions. There are some interesting observations in both cases. People tend to use filler words when describing the objects using speech. For example, the word "like" appears

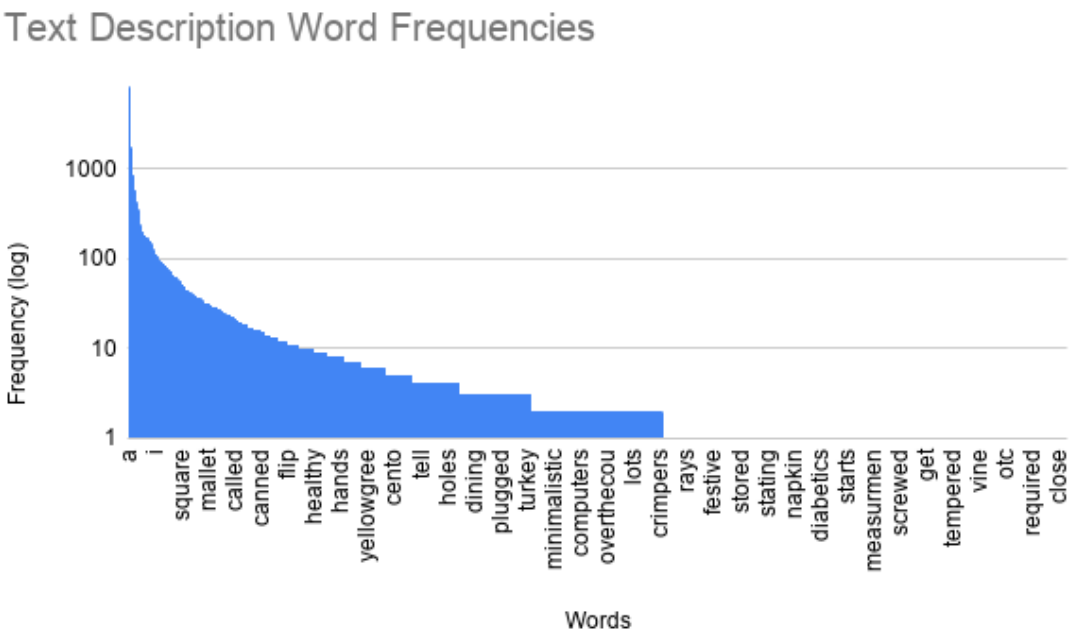


Figure 4.4: Word frequencies of text descriptions within GLD.

166 times in speech data whereas it was not significant in the text data. The frequency of the word "used" is high in both modes which is typically used to describe the functionality of certain objects. This observation is consistent with anecdotal evidence from reviewing the descriptions.

Since the basic hypotheses were refuted, a deeper analysis using more sophisticated methods such as language modeling will be needed to find significant differences between the two domains. More complex differences may include sentence structure and readability. Additionally, I may need more data and demographics on those who provided the descriptions. A child performing the same task would certainly have a less varied vocabulary than a college educated adult. I only collected language from assumed native English speakers, however it is common for non-native English speakers to replace words they don't know with synonyms and

Token	Frequency
black	1073
object	924
white	817
blue	784
red	746
bottle	732
yellow	718
small	482
used	449
pair	436
green	432
plastic	341
box	310
silver	265
metal	220
pink	219
picture	188
orange	174
large	173
jar	164

Token	Frequency
black	599
white	545
blue	427
bottle	385
red	360
yellow	353
object	268
green	231
used	223
handle	210
small	185
color	171
like	166
box	163
silver	163
pair	153
plastic	151
looks	131
pink	109
light	102

Table 4.1: The top 20 most frequent words and their frequencies in the textual descriptions (left) and in the transcribed speech data (right). These distributions are not normalized and there are about twice as many text as speech descriptions.

phrases within their vocabulary.

Chapter 5: Experiment

In this chapter I describe a baseline grounded language acquisition training scenario utilizing my collected dataset.

5.1 Grounded Language Model using Manifold Alignment

For the transfer learning experiments I use two neural networks, trained together using manifold alignment with triplet loss. Manifold alignment is a domain transfer learning technique where two high dimensional spaces, in my case language and vision, are embedded in a lower dimensional space [12, 101, 102]. The key idea is that each neural network transforms related data points from each high dimensional domain to be close together in the lower dimensional domain. Therefore, if a novel pair of data points are sent through the networks and they are close together in the embedded dimension space, then the two are related. Conversely, if they are far apart in the embedded dimension space, then they are unrelated.

In this thesis the two higher dimensional spaces are vision, RGB and depth, and language, textual descriptions of the RGB images. By aligning these two domains the model learns associations between the visual percepts and language. Thus learning how closely the language describes the image, not just a label for what is

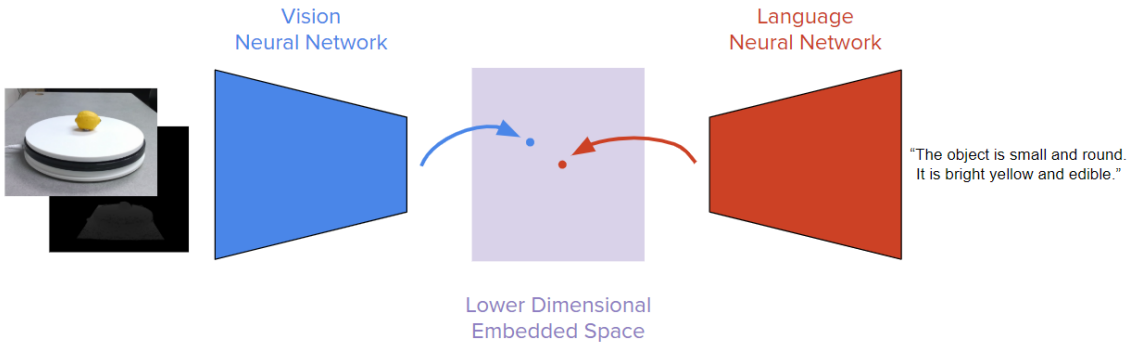


Figure 5.1: A high level view of manifold alignment. The vision and language domains are embedded into a lower dimensional space. Related pairs of vision and text are aligned to be closer to each other within the embedded space. Novel pairs can then be embedded to determine correspondence. Alternatively, inputs from either domain can be embedded in the intermediate space to find associated instances from the other domain.

in the image. I use triplet loss (Equation 5.1) as the loss function for the manifold alignment. Triplet loss is a special loss function that takes as input an anchor, a data point positively associated with the anchor, and a data point negatively associated with the anchor. The loss function then encourages the network to pull the anchor and positive closer together in the embedded space, while repelling the anchor and the negative [13, 103]. The model is trained with triplets of vision, language, and two special triplets where the anchor is vision or language and the positive and negatives are from the other domain.

$$L(A, P, N) = max(\| f(A) - f(P) \|^2 - \| f(A) - f(N) \|^2 + \alpha, 0) \qquad (5.1)$$

5.2 Feature Extraction

Here I describe the process for feature extraction of the raw data.

5.2.1 Language Features

The language features were extracted using BERT, which stands for Bidirectional Encoder Representations from Transformers [55]. BERT is a pretrained English language model that performs well on many benchmark NLP tasks including sentiment analysis, question answering, and inference [48, 104–106].

For each instance in the two datasets, UW and GLD, all ten of its text descriptions were gathered into a single multi-sentence *descriptive document*. BERT's English only lower cased pre-trained model was used to convert each of these documents into a feature vector. The default mean pooling operations was used since each of these documents is a multi-sentence input. This operation averages the word embeddings of all tokens within the document into a single feature vector. This vector is of length 3027 which is the resultant language dimension of the manifold alignment model. Notably, this dimension is the same for both the UW and GLD datasets which allows me to feed data from one dataset into a model trained on the other dataset. This is an important feature that eases transfer learning since I do not need to add another layer to modify the size of the input to the network.

Speech descriptions were not included in the experiments for this thesis. I discuss this further as part of Future Work in Section 6.1.

5.2.1.1 Negative Examples

Many machine learning algorithms, including triplet loss [103], require a concept of a negative example. For many tasks this is a concrete concept. For example, if I have a model that is meant to recognize images of cats, then either the image contains a cat or it doesn't. However, in language, just because a description omits certain words doesn't imply the missing words constitute the opposite or a negative example of that description. For example, someone may describe a bell pepper as "a red, crisp, and fresh vegetable", but it would be a mistake to assume that "green, and gross tasting" was a negative description of a bell pepper class since someone who does not like the green variety could still be describing a bell pepper.

To solve this problem, I follow the work of Pillai and Matuszek [98, 107]. I find the cosine similarity metric between an instance's language feature vector and all of the other language feature vectors within the corpus. Since vectors that are semantically similar will also be similar in their cosine similarity, I can choose from the vectors that are most dissimilar as negative examples. In their work, Pillai and Matuszek found an empirical threshold that gave the best results for negative examples. Similarly, I chose to randomly select from the ten most negatively associated language vectors as a description's "negative example".

5.2.2 Visual Features

Following the work of Eitel et al. [18, 76], I transform and combine the RGB and depth percepts of each dataset into a multimodal visual feature vector using two

CNNs. The RGB and depth channels of each image are run through separate CNNs which are pretrained on the ImageNet dataset and fine tuned on the UW RGB-D dataset [11, 19]. The depth CNN is fine tuned by converting depth images into false RGB color images, red is encoded as close and blue is encoded as far. The final softmax layers that are used for classification are then chopped off the end of the networks, leaving the new final layers as complex feature encoders. The two output vectors are then concatenated together to form a single multimodal visual feature vector. Similarly to the language features, the visual features are always output in a 4096 dimensional vector, no matter the input size, which facilitates transfer learning.

5.3 Baseline Experiments

Since the goal of this work is to be able to learn a grounded language model for my new dataset with minimal training, I have two baselines. Both models are tested on UMBC GLD, but are trained on UW RGB-D and UMBC GLD, respectively.

First a model is trained on the UW RBG-D data. This model will simulate an expensive-to-train, large dataset, pretrained model since the UW RGB-D dataset contains more images than UMBC GLD. However, as mentioned in Section 2.5.1 the text descriptions associated with each image are tied to the instance names, not the images as in UMBC GLD. Since this model is a stand in for a pretrained model that will be used as the basis for transfer learning, I test it on a held out datset from UMBC GLD. This provides a baseline for the transfer learning experiment

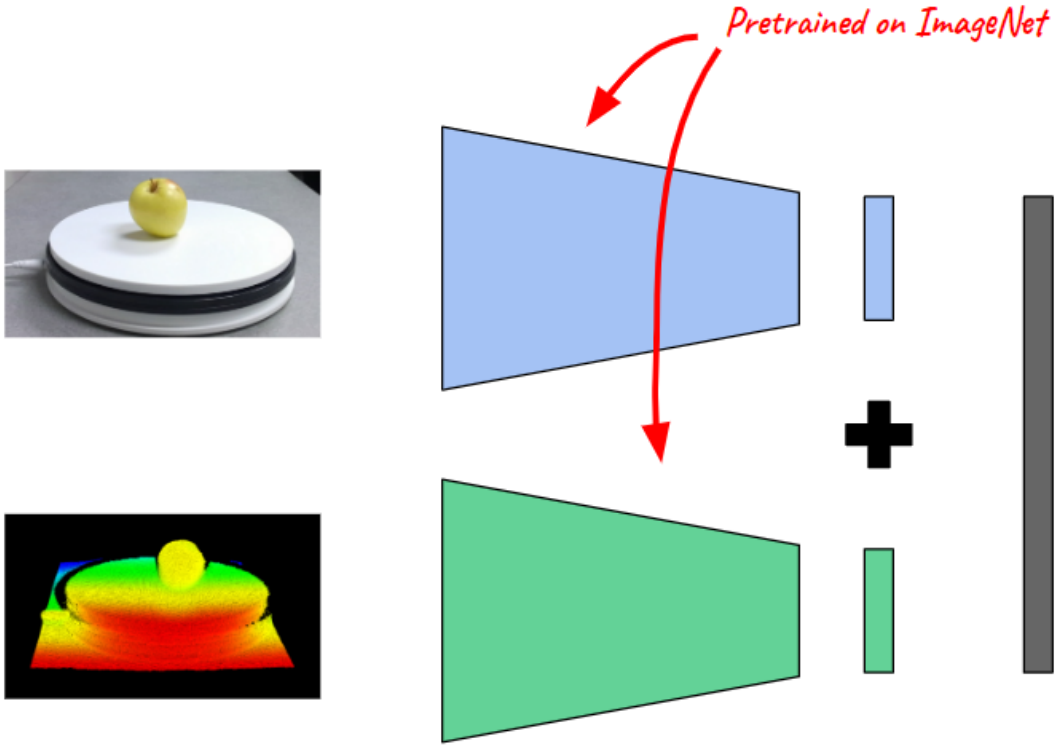


Figure 5.2: Vision feature extraction architecture. Two networks pretrained on ImageNet have their classification layer removed and their final feature layers concatenated together to form a single 4096 dimensional feature vector.

Experiment Name	Training Set	Testing Set	Hypothesis
Baseline 0	UW RGB-D	UW RGB-D	Perform decently well
Baseline 1	UW RGB-D	UMBC GLD	Perform poorly
Baseline 2	UMBC GLD	UMBC GLD	Better than Baseline 1
Transfer Learning	UW RGB-D then UMBC GLD	UMBC GLD	Faster to train, better performance than Baseline 2

Table 5.1: A summary of the grounded language experiments in this thesis.

discussed in Section 5.4 by giving me a sense of how well the model was doing before the transfer learning and how much better or worse it does after the fine tuning. A second model is trained on UMBC GLD and tested on the same held out set from the first baseline experiment. This baseline excludes the pretrained model step and so will let me know how well the model could have performed if starting from random. Additionally, I performed a sanity check "Baseline 0" experiment trained and tested on the UW RGB-D data to ensure that the model was working correcly. A summary of the experiments can be seen in Table 5.1.

5.4 Transfer Learning Experiment

I take the model that was pretrained on the UW RBG-D data and run 50 epochs of training on the UMBC GLD training data. As discussed in Section 2.4, this transfer learning step fine tunes the weights in the pretrained model to perform better in less training time than a model trained on UMBC GLD alone. This fine tuned model is then tested on the same held out GLD testing set as the previous models.

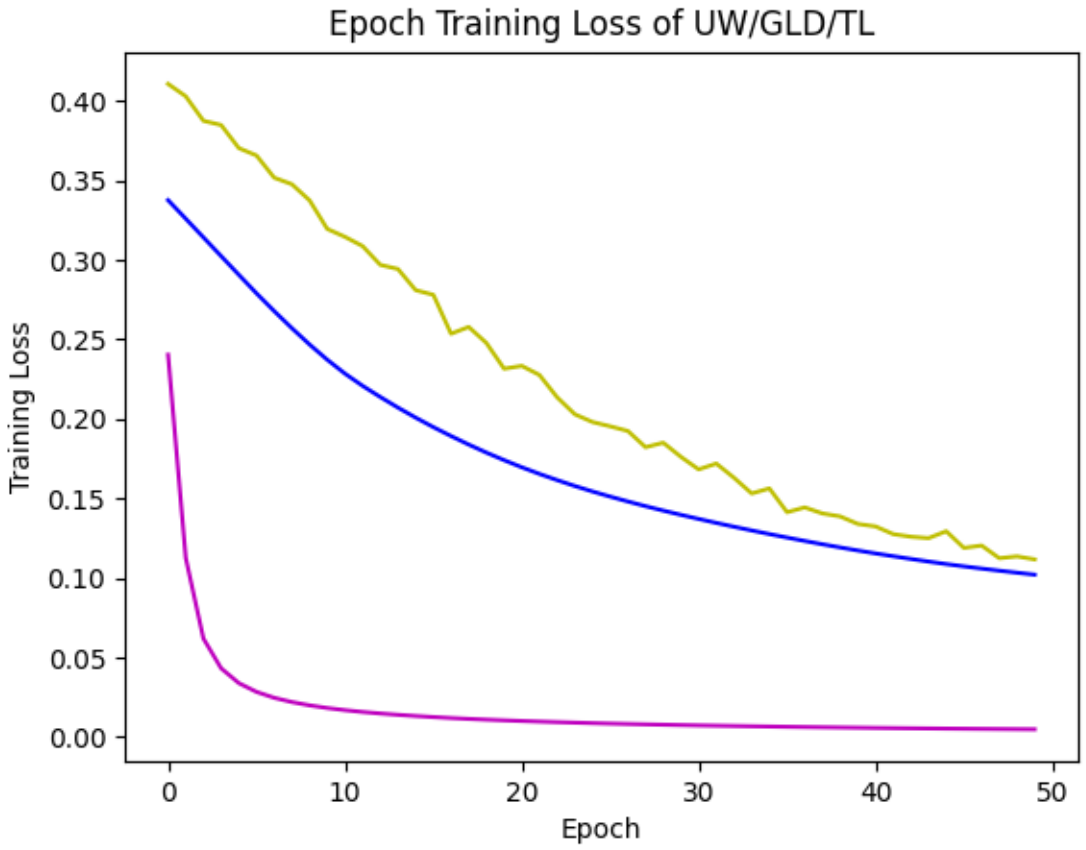


Figure 5.3: Training loss over 50 epochs of a manifold alignment model with triplet loss trained on UW RGB-D data and paired text descriptions. Each epoch includes around 6000 training pairs.

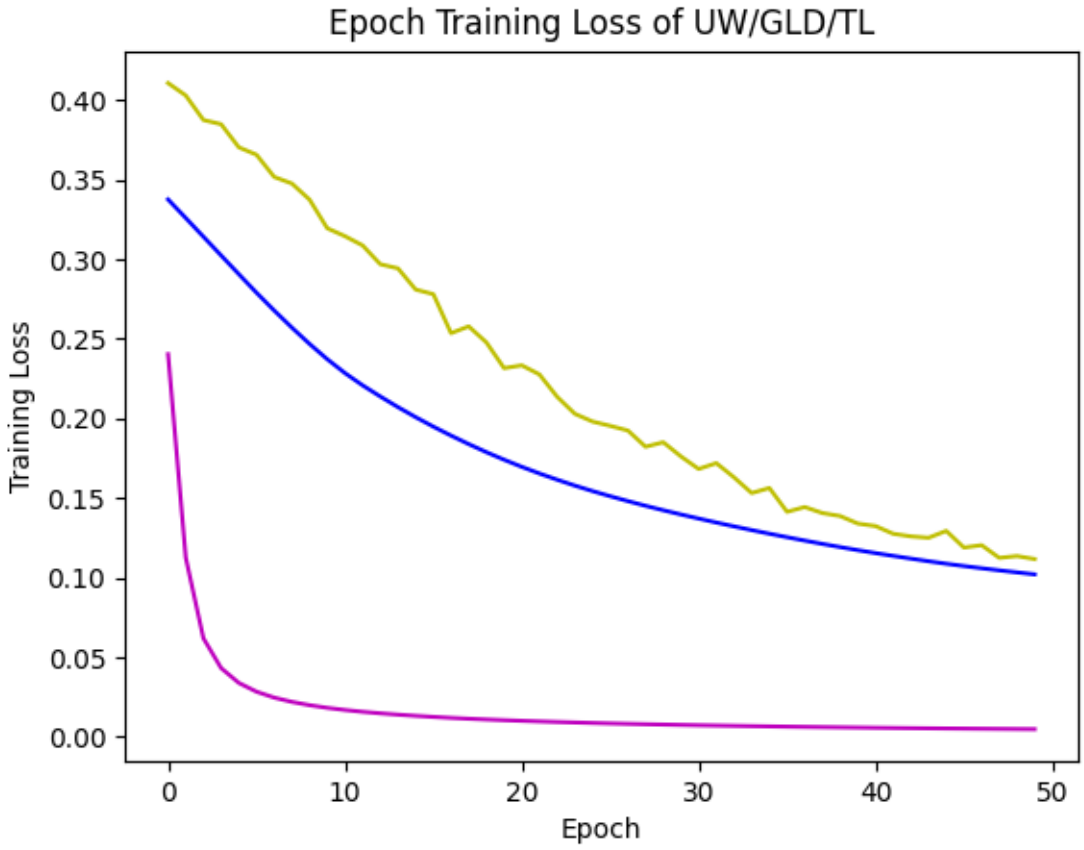


Figure 5.4: Training loss of the GLD baseline experiment and the transfer learning experiment. The GLD baseline was trained on about 600 training examples per epoch for 50 epochs. The transfer learning experiments starts with an initial training loss lower than the baseline experiment due to pretraining on the UW RGB-D data. It then trains faster and achieves a lower training loss by the end of the same 50 epochs as the baseline experiment.

5.5 Within Domain Experiment

One problem that can occur with the manifold alignment model with triplet loss is that while the network may learn to align vision and text, the individual vision and language networks are not consistent within themselves. In other words, pairs of vision and language will be close to each other in the embedded space, but positively associated images will be far apart in the embedded space. I would like not only the pairs of vision and language to be close together, but related images or related language as well.

To test that my model is achieving this, I find pairs of vision and language that are both positively associated and negatively associated to a target pair through the language domain using the technique described in Section 5.2.1.1. If the networks are consistently aligned, then the distance in the embedded space between the target and positively associated instance should be less than the distance between the target and negatively associated instance for both domains. The relevant metric is then how many pairs of the testing set satisfy Equation 5.2, where $d()$ is a distance metric, specifically cosine similarity for language and the euclidean distance for vision. Table 5.5 contains the results from each experiment.

With the exception of the vision net in the transfer learning experiment, each test does better than a random guess. However, the experiments tested on GLD did far better in the language domain than Baseline 0. This could be a result of differences between the language collected for the UW data and the language in GLD. If there are differences that separate the description feature vectors further to

Experiment	Test Set	Vision %	Language %
Baseline 0	UW RGB-D	53.9%	52.6%
Baseline 1	UMBC GLD	53.2%	56.1%
Baseline 2	UMBC GLD	55.5%	73.4%
Transfer Learning	UMBC GLD	48.6%	62.4%

Table 5.2: Results from the within domain experiment, given as the percentage of test samples that satisfy Equation 5.2. Euclidean distance and cosine similarity were used as the distance metrics for vision and language respectively.

begin with, then the network can start from a more distinguishing space, resulting in higher percentages of separation between positive and negative instances. The vision percentages in general are lower than the language percentages. This could be because the language domain was used to determine what was positively or negatively associated to the target sample. Because of this, two descriptions that appear very similar could have very different or very similar images associated with them. Additionally, all of the vision input data looks very similar, the table and turntable are in every image. For the same reason that the language domain may have been more separated from the beginning, this could have resulted in the vision domain being densely grouped from the beginning, resulting in poorer within domain performance.

$$d\Big(f(A) - f(P)\Big) < d\Big(f(A) - f(N)\Big) \tag{5.2}$$

5.6 Analysis

The hypotheses going into the transfer learning experiment are that the first UW RGB-D to UMBC GLD model will perform the worst since the training and testing datasets are separate. The second model will perform better than the first for the complementary reason that the training and testing sets are from the same data set. Finally, the transfer learning model will outperform both of the baselines. It will outperform the first model since it starts from the end of training of the first baseline and should improve its testing on UMBC GLD if it is fine tuned on GLD. The transfer learning experiment should outperform the second baseline since it starts with pretrained weights that are tuned to the same task but not the same data.

To test my models, I first gather pairs of image and text descriptions from the held out testing dataset. In order to calculate Precision, Recall, and F1, I need a threshold distance in the embedded space to determine what is considered a positive association. To do this, I take the pairs from the training set and find the Euclidean distance between the vision and language feature vectors in the embedded dimension. The mean of these distances plus one standard deviation is used as an upper bound on distances that are considered positive. If the distance between two embedded vectors is greater than this bound, then they are considered not associated. This way I can compute the F1 score of all of my models.

Additionally, I report two other metrics: Mean Reciprocal Rank (MRR) and Distance Correlation (DC). MRR is a measure of order preservation. To calculate

Experiment	F1	MRR	DC
Baseline 0	0.1750	0.0313	0.1690
Baseline 1	**0.0226**	0.0824	0.2780
Baseline 2	**0.1230**	0.0930	0.2480
Transfer Learning	**0.1700**	0.0853	0.2270

Table 5.3: Results from tests of F1, MRR, and DC on the baseline and transfer learning experiments. F1 is displayed in bold since it shows the improvements made through transfer learning.

it, the distances of all pairs of an image and description are measured and ranked. Then the multiplicative inverse, or reciprocal, of the rank of the nearest language description that matches the class of the object from the visual data is found for each image. The MRR is the mean of these multiplicative inverses [108]. MRR is therefore between 0 and 1, with 1 being perfect preservation of the order of closeness in the embedded space. The Distance Correlation metric measures how aligned the vision and language embeddings are aligned. Essentially, if two images in the embedded space are close together, then their paired language embeddings should also be close to each other. This is computed by finding the Pearson correlation between the image distances and the language distances of pairs. The range of this metric is -1 to 1 with -1 not being aligned and 1 being perfectly aligned. The results from the tests are shown in Table 5.3.

The F1 scores for these experiments were underwhelming. Based on Richards et al.'s [109] work with similar datasets and algorithms, I have reason to believe that this method should perform better than it did. However, these experiments are still illuminating for the promise of transfer learning for a grounded learning acquisition task. According to the performance metrics, the hypotheses were true. The UW

RGB-D trained network, when tested on the held out GLD testing set, performed the worst. The GLD baseline performed much better. However, the transfer learned network easily outperformed both.

Since the training set of GLD was much smaller than the UW RGB-D dataset, this is promising for the ability to use pretrained manifold alignment models to then quickly fine tune to more specific environments.

Chapter 6: Conclusion

6.1 Future Work

The transfer learning experiment shows that the UW data and GLD data are close enough in domain to be learned through transfer learning techniques. However, more robust domain transfer learning techniques will be needed to perform similar model tuning between different sensors or different environments [110]. In particular, robots come in many different forms and interaction modes [111]. Some may have depth sensors while others may only have RGB cameras while others still may have completely sensors such as LIDAR or SONAR. As discussed in Section 2.2.5, to deploy robots worldwide there is a need to understand many languages. Future work will need to focus on both the differences in visual domains as well as language to create robust models.

This work talks about algorithms for robots but never actually places a grounded language model on a robot interacting with the physical world. As anyone who has every worked with robots will know, the physical world has a way of exposing flaws and edge cases in designs. These models should be loaded onto robots to stretch their capabilities.

As mentioned in Section 5.2, an obvious area of future work is incorporating

the gathered speech descriptions into the learning tasks. From the quality analysis that was done on the two domains, it was clear that the speech data was dirtier than the text data. However, more analysis would be needed to determine the differences between the two modes and how the data quality affects learning.

Data gathering is never complete. As robots move into more parts of our lives they will need more and more training data. Understanding how the speech-to-text transcriptions can help alleviate some of the costs and barriers to producing higher quality datasets to test new theories and models. Additionally, gather more objects and possibly scenes of domestic living will be necessary to enable placing robots in homes.

6.2 Contributions

In this work I have created a novel data set by combining high fidelity imaging of domestic objects with collected text and speech descriptions of the objects. This dataset will enable other researchers to test their grounded language models without the high costs associated with gathering and analyzing text and speech descriptions.

I used that dataset to show that transfer learning is a viable option for fine tuning grounded language models to specific environments. Specifically, it is feasible to have one model that is pretrained on lots of data and use smaller data sets to fine tune that model. This fine tuning step not only saves training time for the end user, but outperforms training a model on solely the smaller dataset.

It is my hope that the continuation of exploring and applying transfer learning

and domain adaptation techniques to grounded language acquisition will lead to smarter, more useful domestic robots. For many people these robots could be a path to a better and more fulfilling life. The ability to fine tune each one to a user's home and lifestyle is essential so that they can provide the help they were built to provide.

Bibliography

[1] Philipp Beckerle, Gionata Salvietti, Ramazan Unal, Domenico Prattichizzo, Simone Rossi, Claudio Castellini, Sandra Hirche, Satoshi Endo, Heni Ben Amor, Matei Ciocarlie, et al. A human–robot interaction perspective on assistive and rehabilitation robotics. *Frontiers in Neurorobotics*, 11(24):1, 2017.

[2] Cory-Ann Smarr, Tracy L Mitzner, Jenay M Beer, Akanksha Prakash, Tiffany L Chen, Charles C Kemp, and Wendy A Rogers. Domestic robots for older adults: attitudes, preferences, and potential. *International journal of social robotics*, 6(2):229–247, 2014.

[3] Jenay M Beer, Cory-Ann Smarr, Tiffany L Chen, Akanksha Prakash, Tracy L Mitzner, Charles C Kemp, and Wendy A Rogers. The domesticated robot: design guidelines for assisting older adults to age in place. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 335–342. ACM, 2012.

[4] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. *arXiv preprint arXiv:1206.6423*, 2012.

[5] Cynthia Matuszek. Grounded language learning: Where robotics and nlp meet. pages 5687–5691, 07 2018.

[6] Raymond J Mooney. Learning to connect language and perception. 2008.

[7] Joyce Y Chai, Maya Cakmak, and Candace Sidner. Teaching robots new tasks through natural interaction. *Interactive Task Learning: Agents, Robots, and Humans Acquiring New Tasks through Natural Interactions*, 2017.

[8] Deb K. Roy. Learning visually grounded words and syntax for a scene description task. *Computer Speech & Language*, 16:353–385, 2002.

[9] Nisha Pillai, Karan K Budhraja, and Cynthia Matuszek. Improving grounded language acquisition efficiency using interactive labeling. *UMBC Student Collection*, 2016.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[11] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. pages 1817–1824, 05 2011.

[12] Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. pages 1120–1127, 01 2008.

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. pages 815–823, 06 2015.

[14] Robert J Schalkoff. *Digital image processing and computer vision*, volume 286. Wiley New York, 1989.

[15] Yann Le Cun, Ofer Matan, Bernhard Boser, John S Denker, Don Henderson, Richard E Howard, Wayne Hubbard, LD Jacket, and Henry S Baird. Handwritten zip code recognition with multilayer networks. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume 2, pages 35–40. IEEE, 1990.

[16] Yoshua Bengio, Yann LeCun, and Donnie Henderson. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks, and hidden markov models. In *Advances in neural information processing systems*, pages 937–944, 1994.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[18] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin A. Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 681–687, 2015.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[21] Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. 2014.

[22] Yajun Xu, Shogo Arai, Fuyuki Tokuda, and Kazuhiro Kosuge. A convolutional neural network for point cloud instance segmentation in cluttered scene trained by synthetic data without color. *IEEE Access*, 2020.

[23] Jianzhong Yuan, Wujie Zhou, and Ting Luo. Dmfnet: Deep multi-modal fusion network for rgb-d indoor scene segmentation. *IEEE Access*, 7:169350–169358, 2019.

[24] Jose M Alvarez, Theo Gevers, Yann LeCun, and Antonio M Lopez. Road scene segmentation from a single image. In *European Conference on Computer Vision*, pages 376–389. Springer, 2012.

[25] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 601–608. IEEE, 2011.

[26] Mahmoud Al-Ayyoub, Aya Nuseir, Kholoud Alsmearat, Yaser Jararweh, and Brij Gupta. Deep learning for arabic nlp: A survey. *Journal of computational science*, 26:522–531, 2018.

[27] Wahab Khan, Ali Daud, Jamal A Nasir, and Tehmina Amjad. A survey on the state-of-the-art machine learning models in the context of nlp. *Kuwait journal of Science*, 43(4), 2016.

[28] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[29] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.

[30] Alexis Conneau, Holger Schwenk, Loıc Barrault, and Yann Lecun. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*, 2, 2016.

[31] Siddharth Patki, Andrea F Daniele, Matthew R Walter, and Thomas M Howard. Inferring compact representations for efficient natural language understanding of robot instructions. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6926–6933. IEEE, 2019.

[32] Jesse Thomason, Shiqi Zhang, Raymond J Mooney, and Peter Stone. Learning to interpret natural language commands through human-robot dialog. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[33] Huda Khayrallah, Sean Trott, and Jerome Feldman. Natural language for human robot interaction. In *International Conference on Human-Robot Interaction (HRI)*, 2015.

[34] James Pustejovsky and Amber Stubbs. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications. "* O'Reilly Media, Inc.", 2012.

[35] Douglass Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing*, pages 133–140, 1992.

[36] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155. Association for Computational Linguistics, 1992.

[37] Helmut Schmid. Part-of-speech tagging with neural networks. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, page 172–176, USA, 1994. Association for Computational Linguistics.

[38] Thorsten Brants. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics, 2000.

[39] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[40] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

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

[42] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[43] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[44] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[45] Francesca Bertacchini, Eleonora Bilotta, and Pietro Pantano. Shopping with a robotic companion. *Computers in Human Behavior*, 77:382–395, 2017.

[46] Pascale Fung, Dario Bertero, Yan Wan, Anik Dey, Ricky Ho Yin Chan, Farhad Bin Siddique, Yang Yang, Chien-Sheng Wu, and Ruixi Lin. Towards empathetic human-robot interactions. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 173–193. Springer, 2016.

[47] Mason Bretan, Guy Hoffman, and Gil Weinberg. Emotionally expressive dynamic physical behaviors in robots. *International Journal of Human-Computer Studies*, 78:1–16, 2015.

[48] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[49] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[50] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

[51] Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5):1–22, 2017.

[52] DG Maynard and Mark A Greenwood. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *LREC 2014 Proceedings*. ELRA, 2014.

[53] VS Subrahmanian, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. The darpa twitter bot challenge. *Computer*, 49(6):38–46, 2016.

[54] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[55] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[56] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[57] Johanna E. Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2016.

[58] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 1:193–206, 2013.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[60] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[61] Paul Kay and Willett Kempton. What is the sapir-whorf hypothesis? *American anthropologist*, 86(1):65–79, 1984.

[62] W. Köhler, 1929.

[63] Nathan Peiffer-Smadja and Laurent Cohen. The cerebral bases of the bouba-kiki effect. *NeuroImage*, 186:679–689, 2019.

[64] Andrew J Bremner, Serge Caparos, Jules Davidoff, Jan de Fockert, Karina J Linnell, and Charles Spence. "bouba" and "kiki" in namibia? a remote culture make similar shape–sound matches, but different shape–taste matches to westerners. *Cognition*, 126(2):165–172, 2013.

[65] David L Chen, Joohyun Kim, and Raymond J Mooney. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435, 2010.

[66] David L. Chen and Raymond J. Mooney. Learning to sportscast: a test of grounded language acquisition. In *ICML*, 2008.

[67] Caroline Patricia Kery, Cynthia Matuszek, Frank Ferraro, and Timothy Oates. *Esta Es Una Naranja Atractiva: Adventures in Adapting an English Language Grounding System to Non-English Data*. University of Maryland, Baltimore County, 2019.

[68] Caroline Kery, Francis Ferraro, and Cynthia Matuszek. ¿ es un plátano? exploring the application of a physically grounded language acquisition system to spanish. In *Proceedings of the Combined Workshop on Spatial Language Understanding (SpLU) and Grounded Communication for Robotics (RoboNLP)*, pages 7–17, 2019.

[69] Steven Brose, Douglas Weber, Ben Salatin, Garret Grindle, Hongwu Wang, Juan Vazquez, and Rory Cooper. The role of assistive robotics in the lives of persons with disability. *American journal of physical medicine & rehabilitation / Association of Academic Physiatrists*, 89:509–21, 06 2010.

[70] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. *arXiv preprint arXiv:1912.01734*, 2019.

[71] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[72] Chunmian Lin, Lin Li, Wenting Luo, Kelvin CP Wang, and Jiangang Guo. Transfer learning based traffic sign recognition using inception-v3 model. *Periodica Polytechnica Transportation Engineering*, 47(3):242–250, 2019.

[73] Shuren Zhou, Wenlong Liang, Junguo Li, and Jeong-Uk Kim. Improved vgg model for road traffic sign recognition. *Computers, Materials & Continua*, 57(1):11–24, 2018.

[74] Matt Macmahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, action in route instructions. In *In Proc. of the Nat. Conf. on Artificial Intelligence (AAAI*, pages 1475–1482, 2006.

[75] C. Matuszek, D. Fox, and K. Koscher. Following directions using statistical machine translation. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258, 2010.

[76] Luke E. Richards and Cynthia Matuszek. Learning to understand non-categorical physical language for human-robot interactions. In *Proceedings of the R:SS 2019 workshop on AI and Its Alternatives in Assistive and Collaborative Robotics (RSS: AI+ACR)*, Freiburg, Germnany, June 2019.

[77] Rui Yan, Yiping Song, Cheng-Te Li, Ming Zhang, and Xiaohua Hu. Opportunities or risks to reduce labor in crowdsourcing translation? characterizing cost versus quality via a pagerank-hits hybrid model. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[78] Jeanine Evers. *Kwalitatief interviewen: kunst én kunde.*

[79] Jeanine C Evers. From the past into the future. how technological developments change our ways of data collection, transcription and analysis. In *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, volume 12, 2011.

[80] Ayelet Weizs. How long it really takes to transcribe (accurate) audio, Jul 2019.

[81] Marjorie Skubic, Dennis Perzanowski, Samuel Blisard, Alan Schultz, William Adams, Magda Bugajska, and Derek Brock. Spatial language for human-robot dialogs. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(2):154–167, 2004.

[82] Matt MacMahon and Brian Stankiewicz. Human and automated indoor route instruction following. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 28, 2006.

[83] Emre Ünal, Ozan Arkan Can, and Yücel Yemez. Visually grounded language learning for robot navigation. In *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications*, pages 27–32, 2019.

[84] Stephen Friedman, Hanna Pasula, and Dieter Fox. Voronoi random fields: Extracting topological structure of indoor environments via place labeling. In *IJCAI*, volume 7, pages 2109–2114, 2007.

[85] Anis Koubâa. *Robot Operating System (ROS).* Springer, 2017.

[86] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[87] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *CVPR 2011*, pages 1729–1736, June 2011.

[88] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *Consumer depth cameras for computer vision*, pages 119–137. Springer, 2013.

[89] Mohammad Reza Loghmani, Mirco Planamente, Barbara Caputo, and Markus Vincze. Recurrent convolutional fusion for rgb-d object recognition. *IEEE Robotics and Automation Letters*, 4(3):2878–2885, 2019.

[90] Joseph K Goodman, Cynthia E Cryder, and Amar Cheema. Data collection in a flat world: The strengths and weaknesses of mechanical turk samples. *Journal of Behavioral Decision Making*, 26(3):213–224, 2013.

[91] Adam J Berinsky, Gregory A Huber, and Gabriel S Lenz. Evaluating online labor markets for experimental research: Amazon. com's mechanical turk. *Political analysis*, 20(3):351–368, 2012.

[92] Huichuan Xia, Yang Wang, Yun Huang, and Anuj Shah. " our privacy needs to be protected at all costs" crowd workers' privacy experiences on amazon mechanical turk. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–22, 2017.

[93] Matt Lovett, Saleh Bajaba, Myra Lovett, and Marcia J Simmering. Data quality from crowdsourced surveys: A mixed method inquiry into perceptions of amazon's mechanical turk masters. *Applied Psychology*, 67(2):339–366, 2018.

[94] Ian Lane, Alex Waibel, Matthias Eck, and Kay Rottmann. Tools for collecting speech corpora via mechanical-turk. In *Proceedings of the NAACL HLT*, pages 184–187, 2010.

[95] Kong Aik Lee, Anthony Larcher, Guangsen Wang, Patrick Kenny, Niko Brummer, David van Leeuwen, Hagai Aronowitz, Marcel Kockmann, Carlos Vaquero, Bin Ma, Haizhou Li, Themos Stafylakis, Jahangir Alam, Albert Swart, and Javier Perez. The reddots data collection for speaker recognition. In *15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[96] Rishabh Sachdeva. In preparation: Speech vs. textual data for grounded language learning. Master's thesis, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, 5 2020. In preparation.

[97] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[98] Nisha Pillai and Cynthia Matuszek. Unsupervised selection of negative examples for grounded language learning. In *Proc. of the Thirty-second AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA, February 2018.

[99] Nisha Pillai, Cynthia Matuszek, and Francis Ferraro. Deep learning for category-free grounded language acquisition. In *Proc. of the NAACL Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics (NAACL-SpLU-RoboNLP)*, Minneapolis, MI, USA, June 2019.

[100] Dimitris Mylonas, Matthew Purver, Mehrnoosh Sadrzadeh, Lindsay Macdonald, and Lewis Griffin. The use of english colour terms in big data. 05 2015.

[101] Chang Wang and Sridhar Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, page 1541–1546. AAAI Press, 2011.

[102] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[103] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. pages 119.1–119.11, 01 2016.

[104] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[105] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[106] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[107] Nisha Pillai and Cynthia Matuszek. Identifying negative exemplars in grounded language data sets. *UMBC Student Collection*, 2017.

[108] Nick Craswell. Mean reciprocal rank. *Encyclopedia of database systems*, 1703, 2009.

[109] Andre Nguyen, Luke E. Richards, Kasra Darvish, Frank Ferraro, Cynthia Matuszek, and Edward Raff. In submission: Cross-modal manifold alignment for grounded language. In *RSS*, 2020.

[110] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[111] Holly A Yanco and Jill L Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, pages 111–119. sn, 2002.