Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

How Faithful Are Current Vision Language Navigation Models: A Study on R2R and R4R Datasets

Semester Thesis

Philip Schefer

pschefer@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Zhao Meng
Prof. Dr. Roger Wattenhofer

July 23, 2020

Acknowledgements

I wish to acknowledge the support from my supervisor Zhao Meng, who greatly supported me during these 14 weeks. I also would like to thank Prof. Roger Wattenhofer for providing the opportunity to write a semester thesis in this topic. As a last point of gratitude, I would like to thank Stephan Roth from the ITET support for the help with the singularity image and other issues I had along the way.

Abstract

To navigate an agent through a physical environment, it has to process visual and spoken instructions and clues in order to reach the goal destination. This often implies that the given information is in no way complete and missing information has to be read from context. Adding to this problem is, as almost always in any machine learning task, the problem of not enough annotated data.

In this project we dive further into the already done work on Vision and Language Navigation. The Speaker-Follower model as well as the environmental dropout model are used in combination with the Room-for-Room (R4R) data augmentation technique. By combining these approaches we hope to see an improvement in generalization, i.e. a higher success rate on unseen environments. Both models are trained and tested on Room-to-Room (R2R) as well as on R4R.

The best results were obtained by the environmental dropout model trained on R4R data. A 32.1 % success rate in unseen environments has been achieved which is a 3.5 % increase considering previous models on the R4R dataset. The navigation error decreased from a previous best of 8.08 m to 7.58 m for our model which is an improvement of 0.5 m.

Chapter 0

Declaration of Originality

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor. For a detailed version of the declaration of originality, please refer to Appendix A.

Contents

Chapter 1

Introduction

General Task

The task of vision and language navigation generally describes an agent (robot) that moves in a physical environment receiving verbal instructions. The agent has to combine the visual and textual input into an action a at any given time t (a_t). A schematic overview of this process is given in Fig. 1.1.

A widely studied topic currently in Visual and Language Navigation is how can one make sure that a given agent really follows the visual and the textual commands it receives. The agent should be capable of following these commands in a complex real world environment. An example of a textual command would be: "Walk past the dining room table and the television. Make a sharp turn left through the doorway and stop at the bottom of the stairs."

In the best case it is able to relate this language instruction perfectly to the visual environment. To study whether the agent really follows these textual instructions and how it follows them, the Room-to-Room (R2R) [1] simulator is used. It uses real world images from the Matterport3D indoor environment [2]. With all this information, namely the complex environments and the human-spoken instructions, the task that we face is composed of problems in vision, language and robotics. The main question now is: How can we improve the performance of the current models and how can we make sure that the agent really follows the spoken instructions?

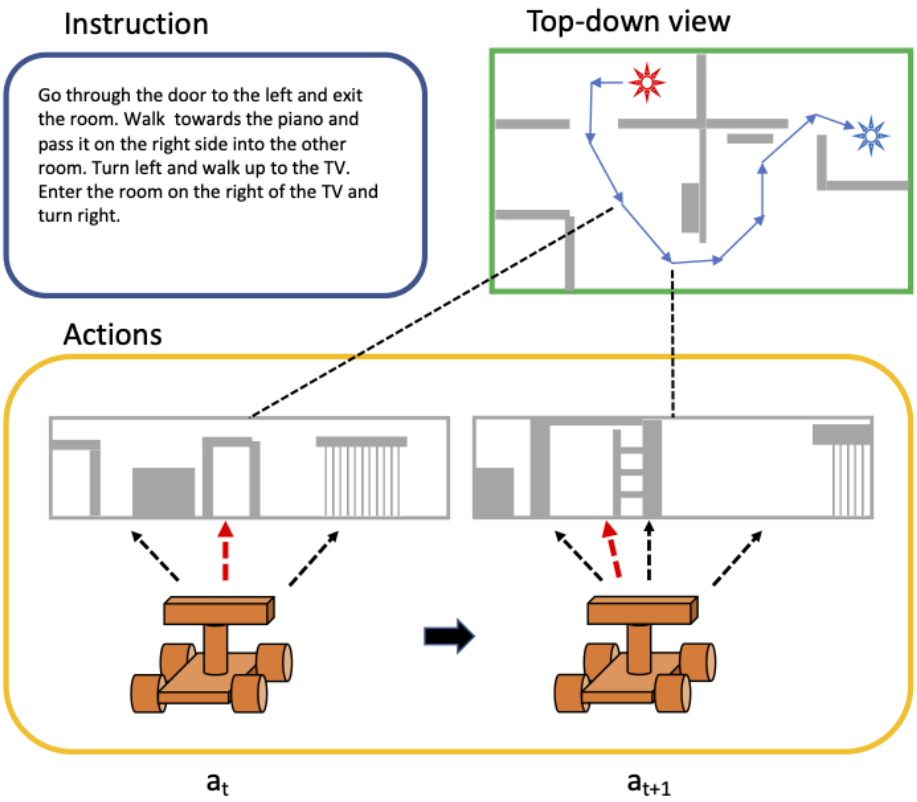

Figure 1.1: General Schematic of the room-to-room task. The orange robot agent receives a spoken instruction and a panoramic action space as input. With this input it has to decide which action at time t (a_t) it will take. The decided action is depicted by the red arrow. It starts from the start point (red star) and tries to reach the end point (blue star). When the robot receives the spoken instruction "stop" and is within a certain threshold distance of the goal location, it successfully solved the task.

Possible Solutions

One of the main points to improve performance in the R2R paper [3] is to use environmental dropout. This strategy is used to counteract the rather poor performance of the R2R model in unseen environments. Speaking of poor performance in unseen environments, the generalization ability of such an algorithm is of course one of the main concerns when developing a navigational robot.

The strategy that Tan *et al.* [3] used to combat the weak generalization ability was to mix imitation learning (IL) and reinforcement learning (RL). This is combined with semi-supervised learning ,i.e. environmental dropout. The last mentioned environmental dropout helps in generating more diverse seen environments, which result in a much better performance of the agent in unseen environments.

Another common problem with training a machine learning model is the lack of training data. The same problem arises in this case where there are not enough "seen" environments to train the agent. Environmental dropout addresses that problem by generating new environments by using dropout of certain visual features. With these new environments generated, a neural speaker model generates new instructions and this data is finally used to train the model. This is close to the approach that Fried *et al.* [4] used, albeit they did not have environmental dropout as an additional tool.

The Speaker-Follower model from Fried *et al.*[4] incorporates the speaker at training and at test time. The speaker synthesizes new routes which it learned from human annotators. At training time, the speaker helps the follower (agent) by synthesizing additional route-instruction pairs. With these labeled pairs, more labeled data is available, which is always a good thing to have when training a machine learning model.

Another major problem in R2R is that paths are mostly shortest paths to the goal location. The metrics used to evaluate the model are also mostly based on goal completion. This however means that the path does not match the spoken instruction a lot of times. To address this the approach from Jain *et al.* [5] is used. In this paper they proposed what they call the Room-for-Room (R4R) approach. R4R is basically the concatenation of paths from R2R to create longer, twistier paths and train the agent on these paths. In these longer, twistier paths the importance of not always going directly to the goal becomes much clearer compared the standard R2R task.

This R4R path conversion emphasizes the impact of language on the navigational agent. As Thomason *et al.* (2019) [6] showed, withholding language inputs from the action sampling agent noticeably reduces the performance in unseen environments. It has been shown that withdrawing language input results in a performance decrease but what about withdrawing visual input. Hu *et al.* [7] looked into this by training a state-of-the-art model without any visual input. They found that models with visual features failed to learn generalizable visual grounding. Even more surprising, they found that models trained without any visual features performed as good as other models on unseen environments. This led the authors to a "mixture-of-experts" approach where they trained the models separately. They train the model once with visual input and once without and combine them afterwards. The resulting model gives them an increased success rate in unseen environments.

Chapter 2

Related Work

There have been a lot of proposed approaches to solving the VLN task in the previous years. All the work done up to this date can be separated into two main groups:

Instruction-based Navigation

The first being the approach of following natural language navigational instructions in an environmental context (MacMahon *et al.*, 2006 [8]; Vogel and Jurafsky, 2010 [9]; Chen and Mooney, 2011 [10]; Andreas and Klein, 2015 [11]; Mei *et al.*, 2016 [12]; Fried *et al.*, 2018 [13]; Misra *et al.*, 2018 [14]). In this task, an agent is required to navigate through an environment given a textual instruction. Among all this work Anderson *et al.* [1] tried something new and introduced a photo-realistic dataset - Room-to-Room (R2R). All images are real images taken by Matterport3D [2] including natural instruction. In these environments the agents' ability to combine the real world inputs becomes even more crucial as it should be able to do it in the real world anyways. Some work has been done by building on top of the work of Anderson *et al.* [1].

Ma *et al.* [15] proposed a self-monitoring agent consisting of two complementary modules namely visual textual co-grounding and a progress monitor to regularize textual grounding. This progress monitor is also used to predict the progress of the agent in the route and tells the agent which instructions already has been completed. Ma *et al.* introduced a self-monitoring agent with two complementary components: (1) a visual-textual co-grounding module to locate the instruction completed in the past, the next instruction and the next moving direction and (2) the progress monitor to ensure that the instruction fits to the navigation of the agent.

Fried *et al.* [4] introduced a Speaker-Follower model in order to synthesize new instructions for data augmentation. The speaker model learns to generate synthetic routes by training on human instructions. This generates more route-instruction pairs, which also means more annotated data to train the model. This same speaker also pragmatically ranks how the candidate action sequences fit to the instruction.

Tan *et al.* [3] has environmental dropout as its main idea. Environmental dropout means that features are dropped out of the environment in order to generate "different" environments to train on and therefore increasing the number of training environments. Environmental dropout is applied to the back translation model, which will be explained in more detail in a later section. Back translation [16] is a popular semi-supervised learning method which has been well studied in the past (Hoang *et al.*, 2018 [17]; Wang *et al.*, 2018 [18]; Prabhumoye *et al.*, 2018 [19]). In back translation, the model learns two translators - a forward one from source to target sentence and a backward one from target to source sentence. With this backward translator, more source sentences are generated from an external language corpus. The newly acquired pairs of source and target sentences are then used to train the forward translator which results in a performance boost of translation ability.

Embodied Vision-and-Language

The second group represents the work done on vision-based navigation tasks (Mirowski *et al.*, 2017 [20]; Zhu *et al.*, 2017 [21]; Yang *et al.*, 2019 [22]; Mirowski *et al.*, 2018 [23]).

Mirowski *et al.* [20] formulated the navigation task as a reinforcement learning problem. In particular they considered jointly learning the goal-driven reinforcement learning problem with auxiliary depth prediction and loop closure classification tasks. With this approach they are able to navigate only using raw sensory input even when the goal location changes frequently.

Zhu *et al.* [21] proposed an agent which can plan a sequence of actions ahead to achieve it's goal. In this approach the agent has to have knowledge about objects and their features, as well as actions and their preconditions and effects.

Yu, Fried *et al.* [7] investigated into the role the vision has on the navigation task. They showed that even when vision is withdrawn from the agent it can still perform really well for several tasks including VLN.

Other approaches

Yu *et al.* [24] tried a different approach considering path sampling. Almost all research uses shortest-path sampling but Yu *et al.* tried Random Walk path sampling to augment the data. By using this approach they managed to close the generalization gap further and achieve better unseen performance in their model.

In a more recent paper Hong *et al.* [25] introduced the notion of splitting up the instructions into smaller chunks. They recognized that many agents only gain little performance by having two streams of information instead of one. Their solution is to provide a fine-grained matching between sub-instructions and the agents visual perception. They propose what they call the Fine-grained R2R dataset or FGR2R. One problem that arises from splitting up the instructions into small parts is that the agents now have problems converting a sub-instruction which depends on a previous sub-instruction. This means when only sub-instructions are used the agents are not capable of seeing connections over the whole instruction.

Chapter 3

Method

3.1 Speaker-Follower Model

The standard model to use in Visual Language Navigation is the instruction-follower model. In this model the agent follows a given instruction and tries to successfully reach the goal. The instructions are human made which means that the number of instructions is limited as it is the case for a lot of supervised learning tasks. To address this problem of not enough labeled paths, Fried *et al.* [4] proposed a Speaker-Follower Model in which they used a speaker model. The speaker model learns to generate route descriptions following a human example. It is first trained on the available ground truth navigation routes and instructions. Before training the follower, the speaker is now able to generate new synthetic routes to add to the already existing physical routes.
The speaker supports the follower both at training as well as at test time. During testing the follower creates specific routes resulting from the input instructions. The speaker then pragmatically ranks these routes. By ranking these routes they have created a pragmatic follower model which means the follower only chooses routes that provide a good explanation of the observed description. This creates a larger amount of supervision data which results in higher generalization ability of the agent when considering unseen environments.
A schematic overview of the model can be seen in Fig. 3.1.

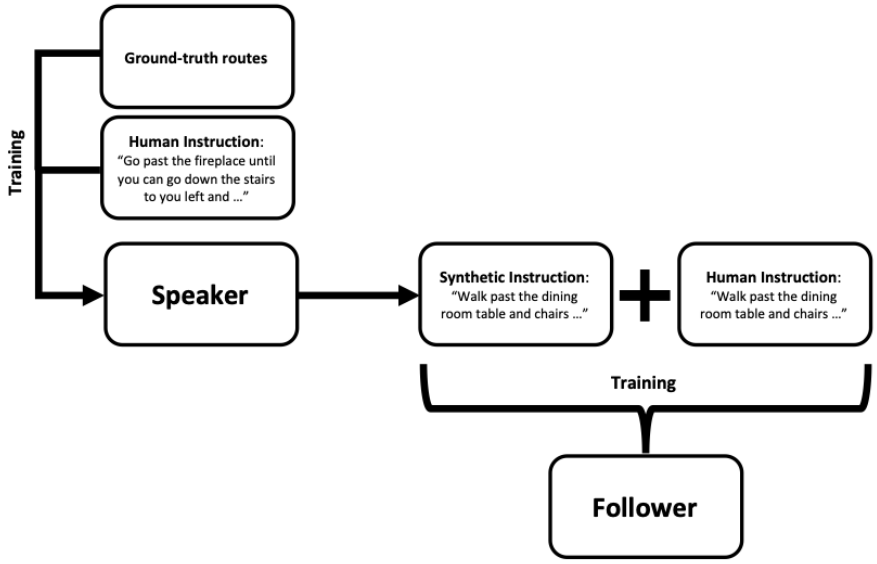

Figure 3.1: Schematic description of the Speaker-Follower model from Fried *et al.*. The speaker is first trained with a combination of ground-truth routes and human made instructions. As an output of the speaker we get a synthetically generated instruction. These synthetic instructions combined with human made instructions are then used to train the follower.

3.1.1 Data Augmentation

To solve the ever existing problem of generalization in machine learning, a lot of data augmentation techniques are used. Fried *et al.* use the synthetic speaker model as mentioned before to increase the number of navigation instructions and route pairs. A human like instruction is generated for each original instruction from the training set using the shortest-path approach. The synthetic data is combined with the original training data and the follower is trained on this combined data set. After the first training the follower is fine tuned with human annotated data.

3.1.2 Panoramic Action Space

The used sequence-to-sequence model in this project has access to a panoramic action space. This means that the agent not just only receives frontal visual sensory input but first builds a 360 degree view of its surrounding. The current viewpoint is discretized into 12 headings of 30 degrees times 3 elevations with 30 degrees. This panoramic action space eliminates the step of first turning the agent in a certain direction to find a feature in the room. The agent can find any feature in its panoramic view in one step, whereas in earlier work [1] it only had a 60 degree frontal window.

3.2 Environmental Dropout

Dropout in neural networks / machine learning applications usually implies that single "neurons" or nodes are dropped out during the training phase. Tan *et al.* [3] proposed a different approach in their paper. They introduced the notion of environmental dropout, which means dropping out real world object from RGB images. When dropping out objects from an RGB image one quickly runs into the problem of being inconsistent over different viewpoints. Tan *et al.* solve this problem by only dropping out the image feature while fixing the orientation feature which keeps connectivity of viewpoints intact.
The forward model in environmental dropout is a navigational agent whereas the backward model is a speaker model similar to the one of the Speaker-Follower. The speaker model used by Tan *et al.* is an enhanced version concretely being a Long-Short-Term-Memory Recurrent-Neural-Network with attention flow. Back translation will be explained in detail in Sec. 3.4.3. This combination of techniques allows Tan *et al.* to generate more data as well as more environments which both lead to a better generalization ability of the model.

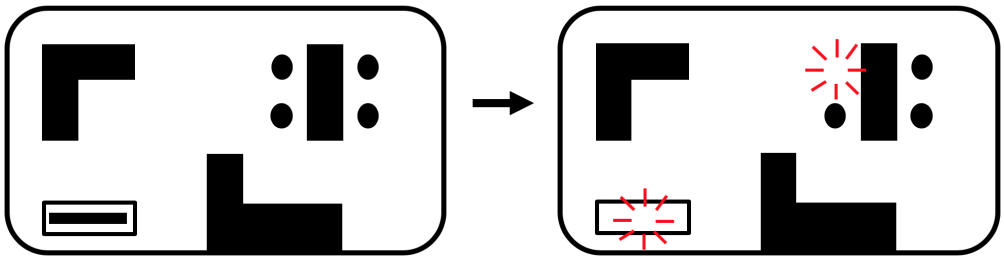

Figure 3.2: Schematic description of the Environmental Dropout model from Tan *et al.*. Objects as a whole get "deleted" from the indoor environments as depicted by the red star shapes.

3.3 Datasets: Difference between R2R and R4R

3.3.1 Room-to-Room (R2R)

In R2R, the environment is represented by a graph where the nodes represent a position the agent can be in. Edges between the nodes indicate that a direct path between two nodes exists. In each node, the agent has an egocentric panoramic view with the images all being captured from indoor environments. The paths paired with language instructions are composed by sequences of nodes in this graph. A path taken in R2R is always the shortest path between two nodes provided it is no shorter than 5m and contains between 4 and 6 edges. Each path has 3 associated language instructions. The average length amount of words per instruction is 29 and the vocabulary consists of 3.1k words in total.

As mentioned before R2R always chooses the shortest path. This however poses a problem since the shortest path may not always correspond to the path that should be taken according to the given instructions. The shortest path has some constraints mentioned above but they cannot solve this problem. A possible solution is given in the next section.

The data is partitioned into one training set, two validation sets and one test set. Different strategies [4] [26] have been used to guide the agent and to introduce intrinsic/extrinsic reward. Fried *et al.* [4] use a follower model which is trained by the student forcing method. That means that the agent's decisions are supervised by the action which takes it closest to the goal. The follower also generates paths which are then scored by the speaker model during inference. The same speaker is used to create an augmented dataset which is used as a training extension to the original R2R dataset.

Wang *et al.* [26] use policy gradients to train their agent. The agent gets a positive reward if it gets closer to the goal and a negative feedback otherwise.

Anderson *et al.* [1] also showed weaknesses in the currently used metrics. They proposed the Success weighted by Path Length (SPL) score which penalizes agents for taking longer paths. This score of course falls apart when we use the R4R data set as explained in the next section. We will also explain the SPL in more detail in chapter 4.1.

3.3.2 Room-for-Room (R4R)

The navigation simulator is based on the Room-to-Room (R2R) simulator [27]. The problem that arises due to the R2R reference paths is that all these paths are shortest-to-goal paths. This in turn means that the agent doesn't have to have a high language conformity in order to obtain a high score. Adding to this is the fact that the largest path in the R2R dataset has only 6 edges. This problem is addressed by the R4R instruction set which generates paths by combining R2R paths together. Existing paths are concatenated if the end node of one path is within a threshold distance of the start node of another path. Figure 3.3 shows a schematic drawing of the R4R path construction.

Each newly generated path will map to $N_A * N_B$ joined instructions, where N_A and N_B are the number of annotations associated with paths A and B.

This approach could of course be taken even further. That means augmenting the R4R data once again by using it as an input with the conversion script from Jain *et al.* [5]. By concatenating the paths from R4R with itself again the resulting paths are even longer and the file size increase considerably. We tried to do it for the training data and ended up with a file exceeding 60 Gb in size. Training on so much data was not possible considering the time available to this project.

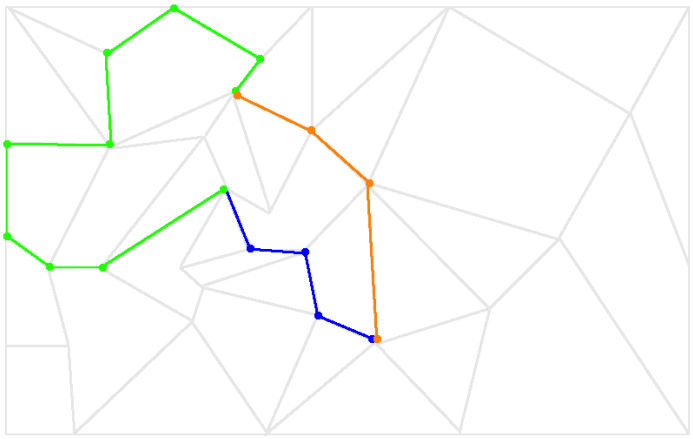

Figure 3.3: This figure shows the concatenation of two paths in a random graph network. The orange path represents the shortest path and the green / blue paths represent two arbitrary R2R paths. The two paths are concatenated to create longer, twistier paths with the same start / end points as the shortest path. The blue path's end node has to be within a threshold distance of 3.0 m of the green starting node in order for the two paths to be concatenated.

3.4 Supervised Learning

3.4.1 Imitation Learning (IL)

When using imitation learning, the agent learns to imitate the behaviour of a teacher. In navigation the teacher selects the next navigable viewpoint which is on the shortest route from the current viewpoint. The agent tries to minimize the negative log probability of the teacher's action.

3.4.2 Reinforcement Learning (RL)

When given a route from IL, the route is the shortest path but that does not mean that it follows the spoken instructions. To make sure that the path taken also follows the textual instructions reinforcement learning is used. In RL the agent learns with certain rewards, for example if the agent stops within a threshold distance from the goal it receives a positive reward and a negative reward otherwise. Tan *et al.* [3] used both IL and RL combined in order to benefit from both learners.

3.4.3 Back-translation

Back translation [16] is a very popular semi-supervised learning method which has been used many times in research up to this date. In the back-translation method the model learns two translators in the first step. The model consists of given source and target sentences and it learns a forward translator from source to target and a backward translator in the other direction. Then it generates more source sentences using the backward translator which generates more data to train the model. Fried *et al.* [4] used this approach when implementing their speaker model to generate a larger amount of synthetically generated language instructions.

Chapter 4

Experimental Setup

4.1 Evaluation Metrics

Path Length (PL) is equal to the total length of the predicted path. It is of course optimal when it's equal to the length of the reference path.

Navigation Error (NE) measures the distance between the last predicted path node and the last reference path node.

Success Rate (SR) shows how many times the last node of the predicted path is within a certain threshold distance of the last reference path node.

Oracle Navigation Error (ONE) measures the smallest distance from any node in the path to the reference goal node.

Oracle Success Rate (OSR) measures how often any node in the path is within a certain threshold distance from the goal. The goal is represented by the last node of the reference path.

Success weighted by Path Length (SPL) [1] takes into account both Success Rate and path length. What it does not consider is the similarity between the intermediate nodes of the predicted path and the reference path. This poses the problem that even though it displays a high score that predicted path did not really follow the spoken instructions but only got the goal right.

The metrics used in this project are the Success Rate, Navigation Error, Path Length and the Success weighted by Path Length.

4.2 Training and Testing

Training and testing is done in a variety of combinations in order to find the one where the overall score benefits the most. The combinations are as follows:

Speaker-Follower model

- Trained on R2R, tested on R2R
- Trained on R2R, tested on R4R
- Trained on R4R, tested on R2R
- Trained on R4R, tested on R4R

and done respectively for the Environmental Dropout model.

For the Environmental Dropout model the speaker as well as the agent are trained for 80'000 iterations. Then the agent is finetuned for another 200'000 iterations starting from the previously achieved best value in unseen environments. When training the speaker and agent, features are dropped out as much as possible. The model is trained with the combination of augmented data and training data.

In the Speaker-Follower model the speaker is first trained with human instructions for a maximum of 20'000 iterations. After the speaker has learned to generate instructions a script is run which actually generates the instructions and stores them in a JSON file. The agent is then trained with this augmented data for 50'000 iterations and fine tuned further for another 20'000 iterations on a combination of original and synthetically generated speaker data.

4.3 Dataset and Simulator

Both agents (Speaker-Follower and Environmental Dropout) are evaluated on the Matterport3D Simulator [1]. The R2R vision-and-language navigation dataset is used as a baseline. In this task the agent receives a spoken instruction which describes the path the agent should follow. The agent then takes multiple discrete actions like turning and moving until it reaches the - what it think is - the goal location. At this location it executes a "stop" action and ends the task. From this dataset the R4R dataset is generated according to the conversion approach from [5].

4.3.1 Data on R2R and R4R

		#samples	PL(mean)	SPD(mean)
R2R	Train	14039	9.91	9.91
	Val. seen	1021	10.2	10.2
	Val. unseen	2249	9.50	9.50
R4R	Train	233532	20.6	10.5
	Val. seen	1035	20.4	11.1
	Val. unseen	45234	20.2	10.1

Table 4.1: Data on the R4R path conversion. PL(mean) denotes the average path lengths and SPD(mean) denotes the average length of the shortest to goal paths.

4.4 Agent

4.4.1 Base Agent Model

Speaker-Follower

The model follows the approach of [1]. It takes the output from the final convolutional layer of a ResNet [28] trained on the ImageNet [29] classification dataset. To increase the ability to generalize, GloVe embeddings [30] are used to initialize the word-embedding vectors in the speaker and follower. The training of the follower consists of student-forcing (sampling actions from the model during training and supervising using the shortest-path to reach the goal). Standard maximum likelihood training with a cross-entropy loss is used to train the base speaker model.

Environmental Dropout

The agent used is based on the encoder-decoder model from Fried *et al.* 2018 [7]. The encoder is a bidirectional Long Short Term Memory (LSTM) - Recurrent Neural network (RNN) with an embedding layer. The decoder of the agent is a regular attentive LSTM-RNN.

The view feature observed in the new environment is calculated by an element-wise multiplication of the original feature and the environmental dropout mask. To maintain the spacial structure, i.e. maintain consistency across different viewpoints, only the ResNet [28] image feature is dropped while the orientation feature is fixed. For a more theoretical view on this please refer to the paper of Tan *et al.* [3].

This approach leads to an improvement of 3 points of the BLEU-4 score [3]. To this model, the environmental dropout approach is added to complete the model.

4.4.2 R4R Speaker

The speaker implemented generates longer instructions from given routes by using the R4R construction of paths. That means paths are concatenated with each other. This concatenation will result in a lower score considering the success rate since it is inherently more difficult to follow a path the longer it gets.

Chapter 5

Results

As already mentioned before, the 2 models were trained on both datasets R2R and R4R using all possible combinations.

- **Environmental Dropout**: The speaker of the Environmental Dropout (EnvDrop / ED) model is trained for 80'000 iterations as a first step. Then the agent is trained another 80'000 iterations using a mixture of human annotated data and speaker synthesized data. After this training the agent is trained further during 200'000 iterations with added environmental dropout.
- **Speaker-Follower**: The speaker of the Speaker-Follower (SF) model is trained for 20'000 iterations until it learned to generated somewhat reasonable instructions. After this the agent is trained again with a mixture of human annotated data and synthetically generated speaker data. To finish the training, the agent is fine-tuned on the original data for another 20'000 iterations. After the training process trajectory predictions with pragmatic inference are obtained by using the rational follower script.

5.1 R2R Training

5.1.1 R2R Evaluation

		Validation Seen				Validation Unseen			
#	Model	PL	NE↓	SR↑	SPL↑	PL	NE↓	SR↑	SPL↑
0	Random [5]	10.4	9.82	5.00	3.70	9.32	9.32	5.20	4.00
1	ED Base [3]	11.0	3.99	62.1	59.0	10.7	5.22	52.2	**48.0**
2	**EnvDrop**	10.3	4.07	62.7	**60.0**	10.48	5.71	47.6	44.0
3	SF Base [4]	-	3.08	70.1	-	-	4.83	54.6	-
4	**Speak.-Foll.**	11.8	**3.05**	**71.5**	-	11.8	**4.64**	**55.6**	-

Table 5.1: Results on R2R Validation Seen and Validation Unseen sets using R2R training data. The first column shows the used models. Success Rate (SR) and Success weighted by Path Length (SPL) are reported in percentages, Path Length (PL) and Navigation Error (NE) in meters. The up-arrow after SR and SPL indicates higher is better, whereas the down-arrow after NE indicates that lower is better. ED / EnvDrop stands for Environmental Dropout and SF stands for Speaker-Follower. The best performance is highlighted bold for each metric except the path length. Our models are highlighted bold for better visibility.

In Table 5.1 a comparison is made between the base performance of Tan *et al.* [3] and Fried *et al.* [4] and our performance using the basic R2R training. As can be seen the results match quite good with a certain range of uncertainty. The ED base performance is chosen to be the "Single Run" performance since it is the most general and highly correlated to the agent's performance. The SF base performance is using data augmentation from the speaker, pragmatic inference and a panoramic action space. The Speaker-Follower model number 4 displays a rather high success rate of 71.5 % which can be misleading though. As stated in earlier research [5] the success rate on the R2R dataset does not really give a value on how much the path conforms to the reference path. It only evaluates if the agent reached the goal location within a certain threshold distance. The same is true for the unseen best performance with 55.6 %. Both these scores these scores have been achieved with the standard Speaker-Follower model which is odd since the environmental dropout model is supposed to use an improved speaker-model version.

5.1.2 R4R Evaluation

It is interesting to see in Table 5.2 that the EnvDrop model performed a bit better in seen environments than it's R2R counterpart when evaluated on R4R data. The success rate of the agent in seen environments increased from 62.7% to 63.5%. Although this performance increase is marginal, the expectation would be to see a drop in success rate since it is inherently more difficult to follow a longer path than it is to follow a shorter path. In unseen environments the performance drops as expected from 52.2% success rate to 45.7%.

		Validation Seen				**Validation Unseen**			
#	Model	PL	NE↓	SR↑	SPL↑	PL	NE↓	SR↑	SPL↑
0	Random [5]	21.8	11.4	13.1	2.00	23.6	10.4	13.8	2.20
1	R2R EnvDrop	10.3	4.07	62.7	60.0	10.48	5.71	47.6	**44.0**
2	**EnvDrop**	10.1	3.92	63.5	**61.0**	14.1	5.51	45.7	40.0
3	R2R SF	11.8	**3.05**	**71.5**	-	11.8	**4.64**	**55.6**	-
4	**Speak.-Foll.**	16.4	7.26	20.6	-	16.9	8.77	16.0	-

Table 5.2: Results on R4R Validation Seen and Validation Unseen sets. Success Rate (SR) and Success weighted by Path Length (SPL) are reported in percentages, Path Length (PL) and Navigation Error (NE) in meters. The random agent number 0 is taken from Jain *et al.* [5] for comparison purposes. R2R EnvDrop and R2R SF are the two models evaluated on R2R to enable a direct comparison between the two datasets.

We can see in Tab. 5.2 that even though the Environmental Dropout model was trained on R2R data, it reached a decent score both regarding the Navigation Error as well as the Success Rate. The Speaker-Follower model however did not do so well. The performance drops considerably when tested on R4R data as can be seen in Table 5.2 number 4. The success rate in seen environments drops from 71.5 % to 20.6 % and from 55.6 % to 16.0 % in unseen environments. The navigation error increases by a factor of about 2 in both cases.

5.2 R4R Training

5.2.1 R2R Evaluation

		Validation Seen				Validation Unseen			
#	Model	PL	NE↓	SR↑	SPL↑	PL	NE↓	SR↑	SPL↑
0	Random [5]	10.4	9.82	5.00	3.70	9.32	9.32	5.20	4.00
1	R2R EnvDrop	10.3	4.07	62.7	**60.0**	10.5	5.71	47.6	**44.0**
2	**EnvDrop**	15.4	5.05	50.6	41.0	30.1	7.84	30.9	18.0
3	R2R SF	11.8	**3.05**	**71.5**	-	11.8	**4.64**	**55.6**	-
4	**Speak.-Foll.**	11.7	3.97	60.0	-	11.5	5.43	42.8	-

Table 5.3: Results on R2R Validation Seen and Validation Unseen sets. The first column shows the used models. Success Rate (SR) and Success weighted by Path Length (SPL) are reported in percentages, Path Length (PL) and Navigation Error (NE) in meters. R2R EnvDrop and R2R SF are the two models evaluated on R2R to enable a direct comparison between the two datasets.

In Table 5.3 we see that the average path length of the environmental dropout model increased drastically when training on R4R data, whereas the path length in the Speaker-Follower model stays more or less constant. The average path length of the environmental dropout model increased from 10.48 m to 30.1 m in unseen environments. This increase in average path length is to be expected since R4R consists out of longer paths. Considering the success rate we see the expected drop in performance for both models. The success rate of the EnvDrop model drops from 47.6 % to 30.9 % and the Speaker-Follower model drops from 55.6 % to 42.8 % in unseen environments. The lowest navigation error is still held by the standard R2R Speaker-Follower model with 3.05 m which makes sense since longer paths are inherently more difficult to follow. The R4R trained Speaker-Follower model comes close to this value with 3.97 m of navigation error. The SPL score is of course expected to drop considerably when working with R4R datasets since the paths get much longer. Our R4R trained environmental dropout model reaches an SPL of 18.0 % in unseen environments.

5.2.2 R4R Evaluation

		Validation Seen				Validation Unseen			
#	Model	PL	NE↓	SR↑	SPL↑	PL	NE↓	SR↑	SPL↑
0	Random [5]	21.8	11.4	13.1	2.0	23.6	10.4	13.8	2.2
1	RCM GO [5]	24.5	5.11	55.5	32.3	32.5	8.45	28.6	10.2
2	RCM FO [5]	18.8	5.37	52.6	30.6	28.5	8.08	26.1	7.7
3	**EnvDrop**	23.8	**4.34**	**55.9**	**38.8**	32.8	**7.58**	**32.1**	**15.3**
4	SF [5]	15.4	5.35	51.9	37.3	19.9	8.47	23.8	12.2
5	**Speak.-Foll.**	8.94	5.89	49.6	-	6.79	6.68	24.5	-

Table 5.4: Results on R4R Validation Seen and Validation Unseen sets. The first column shows the different combinations of models and datasets.Success Rate (SR) and Success weighted by Path Length (SPL) are reported in percentages, Path Length (PL) and Navigation Error (NE) in meters. Model number 1 is the goal oriented Reinforced Cross Modal Matching agent evaluated by Jain *et al.* [5] which was originally introduced by Wang *et al.* [26]. This agent is listed because it reached the highest success rate on the R4R dataset in previous research. Model number 2 is the fidelity oriented RCM agent evaluated by Jain *et al.* [5] which reached the previously best navigation error on the R4R dataset. The agent number 3 is the Speaker-Follower model evaluated on R4R from Jain *et al.* [5].

In Table 5.4 it can again be seen that considering the average path length, our Speaker-Follower model stands on its own. Even though the Speaker-Follower model evaluated by Jain *et al.* [5] also did not reach the average path lengths of the other models, our model's average path lengths remains to be the shortest. The path lengths are 8.94 m / 6.79 m for seen / unseen environments respectively. These small average path lengths also explain the navigation error in unseen environments for our Speaker-Follower model. The navigation error is 6.68 m which represents the lowest value among all the models. This however has to be taken with a grain of salt since if the model already operates on shorter paths it is less difficult to reach the goal within a certain distance.
This poses the question of why these average distances are so much smaller than for the other models. The success rate on the other hands is similar to the performance Jain *et al.* received in both seen and unseen environments. It has to be said though that it is not clear if Jain *et al.* used the same Speaker-Follower model from Fried *et al.* [4] that we used to generate predictions.

To finish up this results chapter we present our environmental dropout model (number 3 in the list) which was trained on R4R data as well as evaluated on the same dataset. In seen environments it performs better in every category than any model listed by Jain *et al.* considering the R4R dataset. The navigation error is 4.34 m which displays an improvement despite having to process longer paths. The success rate reached 55.9 % which is only marginally better than the already tested RCM model at 55.5 %.

In unseen environments the success rate of our environmental dropout model reached a value of 32.1 % which is an increase of 3.5 % to the previous best model on the R4R dataset. The navigation error decreases from a previous best of 8.08 m (model number 2) to 7.58 m for our environmental dropout model.

5.3 Path Conformity in R4R

5.3.1 Positive Example

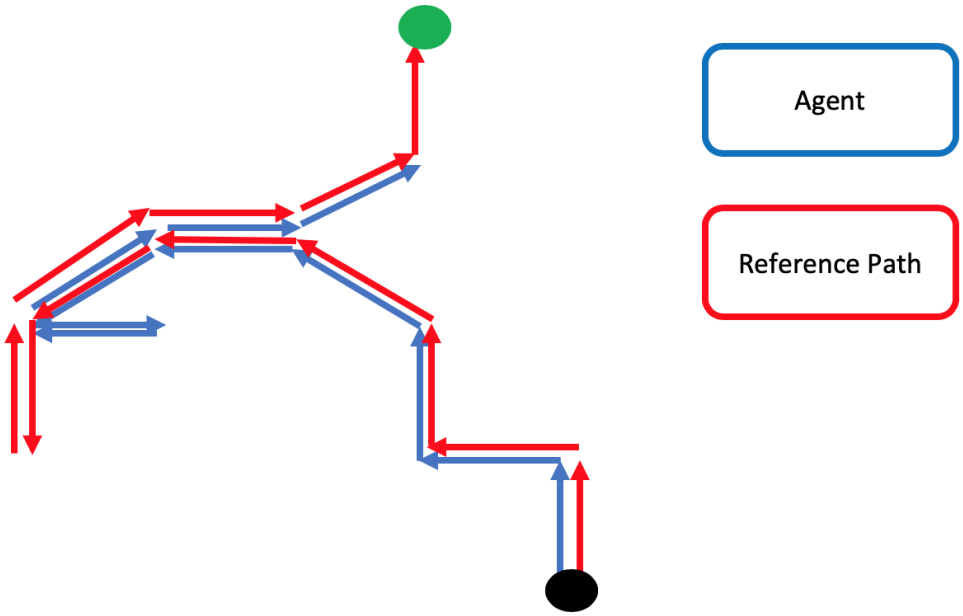

Figure 5.1: This figure shows an agent which conformed good to the given instructions. The black circle is the starting point and the green circle is the goal location. The blue arrows represent the agent and the red arrows represent the Reference path. The navigation error of the agent is 1.84 m. The agent took 10 steps in comparison to the 11 steps of the reference path. A path is labeled as successful if the agent stops within 3 m of the goal location.

The case that the agent almost perfectly follows the instructions as can be seen in Fig. 5.1 is rarely seen in R4R. This is the case because the instructions are long and very difficult to follow. Although the agent does not conform to the reference path in many cases, the amount of shortest-paths taken is reduced in R4R.

5.3.2 Negative Example

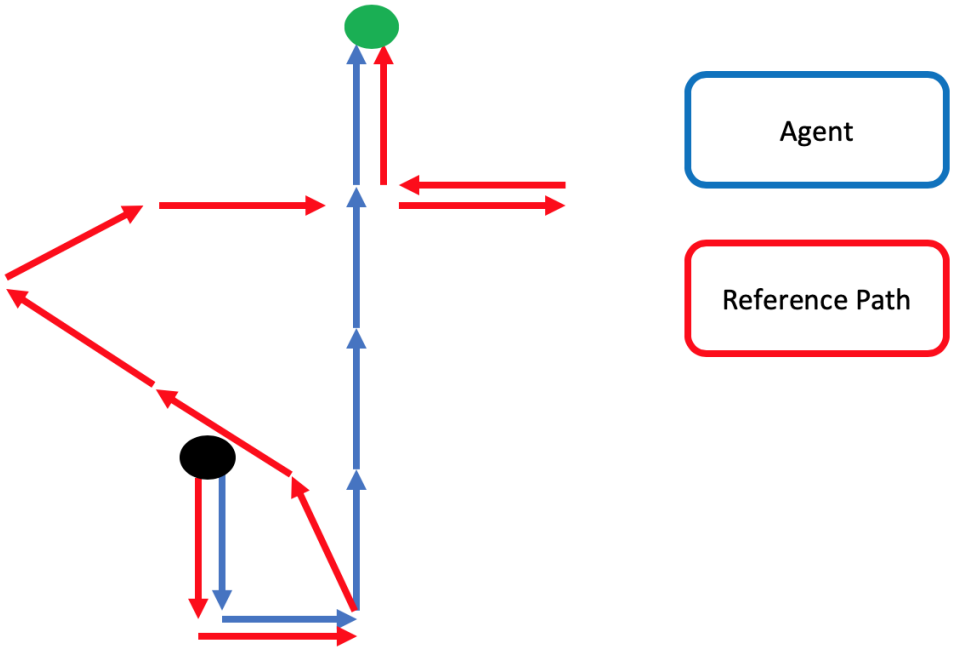

Figure 5.2: In this figure an agent which did not follow the instructions well is shown. The black circle is the starting point and the green circle is the goal location. The blue arrows represent the agent and the red arrows represent the Reference path. The navigation error is 0 m but the agent lacks conformity. The agent took 6 steps, whereas the reference path consists of 10 steps.

Fig. 5.2 displays the case where the agent did not follow the instructions correctly. Even though the navigation error is 0 m, the agent just took the shortest-path to the destination. The reference path is 4 steps longer than the path of the agent. This discrepancy in number of steps could be an additional metric in future research to determine the conformity level of any given agent. This would represent a metric similar to the Coverage weighted by Length Score used by Jain *et al.* [5]. They also take Path Coverage into account and combine it with the Length Score.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Even though the increase in performance of our environmental dropout model combined with the R4R data augmentation technique is not that great, it certainly points into a direction. The direction being to combine different data augmentation techniques and different environmental augmentation techniques. One example used in this project being the environmental dropout combined with the R4R data augmentation. Some combinations also showed that they gain little to no performance when combined. An example being the Speaker-Follower model combined with the R4R data augmentation process.
Using paths from the R4R dataset certainly increases the fidelity of the agent [5] to the path throughout the whole path and gets away from the shortest-path oriented agent.

Our agent closes the generalization gap between seen and unseen environments by another 3.5 % compared to previous work. If this trend is continued and more work is done on the R4R dataset the generalization gap can surely be decreased over time. By reaching a good performance of the agent in unseen environments, it is then able to tackle real world situations and by training it on the R4R dataset we can make sure that it better conforms to the spoken instruction. In a real world environment it is of utmost importance that the robot (agent) conforms to the given instructions. Otherwise the robot could walk into unwanted areas and be damaged or be fully destroyed in the worst case.

Even with this improvement of success rate and navigation error there is still plenty to achieve in the R2R / R4R navigation task, since the human navigation error is only 1.61 m [1].

6.2 Future Work

There are a lot of other different approaches that could have been done to solve the task but the time constraint was a limiting factor in this semester-project. One approach would be to further fine-tune the models on the two datasets interchangeably. That means train a model on R2R which is already trained on R4R and the other way around. This approach leads to the topic of multitask learning where the models could be even be used together to get the best out of each model. Multitask learning could be used in combination with a loss function where the two model each have their own prefactor.

$$Loss = \alpha * Loss_{R2R} + \beta * Loss_{R4R}$$

With alpha and beta as prefactors, one could tune the model exactly to increase the performance.

There has also been work done on different path sampling techniques. Yu *et al.* [24] proposed random walk path sampling which generated better performance in unseen environments. It would be interesting to see how random walk path sampling is able to close the generalization gap for the environmental dropout model in combination with the longer R4R paths.

The newly proposed model of Hong *et al.* [25] could also be used in combination with the R4R dataset. This would increase the path conformity of the agent even more since Hong *et al.* broke up the instructions into smaller sub-instructions to increase the agent's conformity. However it remains unknown whether or not this will also increase the overall performance since this approach is known to have problems with parts of the instructions which are connected with other instruction parts.

Bibliography

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[3] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," *CoRR*, vol. abs/1904.04195, 2019. [Online]. Available: http://arxiv.org/abs/1904.04195

[4] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-language navigation," in *Neural Information Processing Systems (NeurIPS)*, 2018.

[5] V. Jain, G. Magalhães, A. Ku, A. Vaswani, E. Ie, and J. Baldridge, "Stay on the path: Instruction fidelity in vision-and-language navigation," *CoRR*, vol. abs/1905.12255, 2019. [Online]. Available: http://arxiv.org/abs/1905.12255

[6] J. Thomason, D. Gordon, and Y. Bisk, "Shifting the baseline: Single modality performance on visual navigation & QA," *CoRR*, vol. abs/1811.00613, 2018. [Online]. Available: http://arxiv.org/abs/1811.00613

[7] R. Hu, D. Fried, A. Rohrbach, D. Klein, T. Darrell, and K. Saenko, "Are you looking? grounding to multiple modalities in vision-and-language navigation," *CoRR*, vol. abs/1906.00347, 2019. [Online]. Available: http://arxiv.org/abs/1906.00347

[8] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'06. AAAI Press, 2006, p. 1475–1482.

[9] A. Vogel and D. Jurafsky, "Learning to follow navigational directions," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics,

Jul. 2010, pp. 806–814. [Online]. Available: https://www.aclweb.org/anthology/P10-1083

[10] D. L. Chen and R. J. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, ser. AAAI'11. AAAI Press, 2011, p. 859–865.

[11] J. Andreas and D. Klein, "Alignment-based compositional semantics for instruction following," *CoRR*, vol. abs/1508.06491, 2015. [Online]. Available: http://arxiv.org/abs/1508.06491

[12] H. Mei, M. Bansal, and M. R. Walter, "Listen, attend, and walk: Neural mapping of navigational instructions to action sequences," *CoRR*, vol. abs/1506.04089, 2015. [Online]. Available: http://arxiv.org/abs/1506.04089

[13] D. Fried, J. Andreas, and D. Klein, "Unified pragmatic models for generating and following instructions," *CoRR*, vol. abs/1711.04987, 2017. [Online]. Available: http://arxiv.org/abs/1711.04987

[14] D. K. Misra, A. Bennett, V. Blukis, E. Niklasson, M. Shatkhin, and Y. Artzi, "Mapping instructions to actions in 3d environments with visual goal prediction," *CoRR*, vol. abs/1809.00786, 2018. [Online]. Available: http://arxiv.org/abs/1809.00786

[15] C.-Y. Ma, J. Lu, Z. Wu, G. AlRegib, Z. Kira, R. Socher, and C. Xiong, "Self-monitoring navigation agent via auxiliary progress estimation," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1901.03035

[16] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *CoRR*, vol. abs/1511.06709, 2015. [Online]. Available: http://arxiv.org/abs/1511.06709

[17] V. C. D. Hoang, P. Koehn, G. Haffari, and T. Cohn, "Iterative back-translation for neural machine translation," in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 18–24. [Online]. Available: https://www.aclweb.org/anthology/W18-2703

[18] X. Wang, H. Pham, Z. Dai, and G. Neubig, "Switchout: an efficient data augmentation algorithm for neural machine translation," *CoRR*, vol. abs/1808.07512, 2018. [Online]. Available: http://arxiv.org/abs/1808.07512

[19] S. Prabhumoye, Y. Tsvetkov, R. Salakhutdinov, and A. W. Black, "Style transfer through back-translation," *CoRR*, vol. abs/1804.09000, 2018. [Online]. Available: http://arxiv.org/abs/1804.09000

[20] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," *CoRR*, vol. abs/1611.03673, 2016. [Online]. Available: http://arxiv.org/abs/1611.03673

[21] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," *CoRR*, vol. abs/1609.05143, 2016. [Online]. Available: http://arxiv.org/abs/1609.05143

[22] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," *CoRR*, vol. abs/1810.06543, 2018. [Online]. Available: http://arxiv.org/abs/1810.06543

[23] P. Mirowski, M. K. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, K. Kavukcuoglu, A. Zisserman, and R. Hadsell, "Learning to navigate in cities without a map," *CoRR*, vol. abs/1804.00168, 2018. [Online]. Available: http://arxiv.org/abs/1804.00168

[24] F. Yu, Z. Deng, K. Narasimhan, and O. Russakovsky, "Take the scenic route: improving generalization in vision-and-language navigation," in *CVPR Visual Learning with Limited Labels Workshop*, 2020.

[25] Y. Hong, C. Rodríguez, Q. Wu, and S. Gould, "Sub-instruction aware vision-and-language navigation," 04 2020.

[26] L. Wang, Y. Zhu, and H. Pan, "Unsupervised reinforcement learning for video summarization reward function," in *Proceedings of the 2019 International Conference on Image, Video and Signal Processing*, ser. IVSP 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 40–44. [Online]. Available: https://doi.org/10.1145/3317640.3317658

[27] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," *CoRR*, vol. abs/1711.07280, 2017. [Online]. Available: http://arxiv.org/abs/1711.07280

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014. [Online]. Available: http://arxiv.org/abs/1409.0575

[30] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

Appendix A

Declaration of Originality

Appendix B

Github / Weblinks

B.1 Github repositories

$https://github.com/ronghanghu/speaker\_follower$

$https://github.com/airsplay/R2R-EnvDrop$

$https://github.com/google-research/google-research/tree/master/r4r$

Our repositories

$https://github.com/Pschefer/Speaker\_follower\_model$

$https://github.com/Pschefer/Semesterproject\_VLN$