

Multi-modal Discriminative Model for Vision-and-Language Navigation

Haoshuo Huang* Vihan Jain* Harsh Mehta Jason Baldridge Eugene Ie

Google AI Language

{haoshuo, vihan, harshm, jridge, eugeneie}@google.com

Abstract

Vision-and-Language Navigation (VLN) is a natural language grounding task where agents have to interpret natural language instructions in the context of visual scenes in a dynamic environment to achieve prescribed navigation goals. Successful agents must have the ability to parse natural language of varying linguistic styles, ground them in potentially unfamiliar scenes, plan and react with ambiguous environmental feedback. Generalization ability is limited by the amount of human annotated data. In particular, *paired* vision-language sequence data is expensive to collect. We develop a discriminator that evaluates how well an instruction explains a given path in VLN task using multi-modal alignment. Our study reveals that only a small fraction of the high-quality augmented data from Fried et al. (2018), as scored by our discriminator, is useful for training VLN agents with similar performance on previously unseen environments. We also show that a VLN agent warm-started with pre-trained components from the discriminator outperforms the benchmark success rates of 35.5 by 10% relative measure on previously unseen environments.

1 Introduction

There is an increased research interest in the problems containing multiple modalities (Yu and Siskind, 2013; Chen et al., 2015; Vinyals et al., 2017; Harwath et al., 2018). The models trained on such problems learn similar representations for related concepts in different modalities. Model components can be pretrained on datasets with individual modalities, the final system must be trained (or fine-tuned) on task-specific datasets (Girshick et al., 2014; Zeiler and Fergus, 2014).

In this paper, we focus on vision-and-language navigation (VLN), which involves understanding

visual-spatial relations as described in instructions written in natural language. In the past, VLN datasets were built on virtual environments, with MacMahon et al. (2006) being perhaps the most prominent example. More recently, challenging photo-realistic datasets containing instructions for paths in real-world environments have been released (Anderson et al., 2018b; de Vries et al., 2018; Chen et al., 2018). Such datasets require annotations by people who follow and describe paths in the environment. Because the task is quite involved—especially when the paths are longer—obtaining human labeled examples at scale is challenging. For instance, the Touchdown dataset (Chen et al., 2018) has only 9,326 examples of the complete task. Others, such as Cirik et al. (2018) and Hermann et al. (2019) side-step this problem by using formulaic instructions provided by mapping applications. This makes it easy to get instructions at scale. However, since these are not *natural* language instructions, they lack the quasi-regularity, diversity, richness and errors inherent in how people give directions. More importantly, they lack the more interesting connections between language and the visual scenes encountered on a path, such as *head over the train tracks, hang a right just past a cluster of palm trees and stop by the red brick town home with a flag over its door*.

In general, the performance of trained neural models is highly dependent on the amount of available training data. Since human-annotated data is expensive to collect, it is imperative to maximally exploit existing resources to train models that can be used to improve the navigation agents. For instance, to extend the Room-to-Room (R2R) dataset (Anderson et al., 2018b), Fried et al. (2018) created an augmented set of instructions for randomly generated paths in the same underlying environment. These instructions were generated by a speaker model that was trained on the available

*Authors contributed equally.

human-annotated instructions in R2R. Using this augmented data improved the navigation models in the original paper as well as later models such as Wang et al. (2018a). However, our own inspection of the generated instructions revealed that many have little connection between the instructions and the path they were meant to describe, raising questions about what models can and should learn from noisy, automatically generated instructions.

We instead pursue another, high precision strategy for augmenting the data. Having access to an environment provides opportunities for creating instruction-path pairs for modeling alignments. In particular, given a path and a navigation instruction created by a person, it is easy to create incorrect paths by creating permutations of the original path. For example, we can hold the instructions fixed, but reverse or shuffle the sequence of perceptual inputs, or sample random paths, including those that share the start or end points of the original one. Crucially, given the diversity and relative uniqueness of the properties of different rooms and the trajectories of different paths, it is highly unlikely that the original instruction will correspond well to the mined negative paths.

This negative path mining strategy stands in stark contrast with approaches that create new instructions. Though they cannot be used to directly train navigation agents, negative paths can instead be used to train discriminative models that can assess the fit of an instruction and a path. As such, they can be used to judge the quality of machine-generated extensions to VLN datasets and possibly reject bad instruction-path pairs. More importantly, the components of discriminative models can be used for initializing navigation models themselves and thus allow them to make more effective use of the limited positive paths available.

We present four main contributions. First, we propose a *discriminator* model (Figure 1) that can predict how well a given instruction explains the paired path. We list several cheap negative sampling techniques to make the discriminator more robust. Second, we show that only a small portion of the augmented data in Fried et al. (2018) are high fidelity. Including just a small fraction of them in training is sufficient for reaping most of the gains afforded by the full augmentation set: using just the top 1% augmented data samples, as scored by the discriminator, is sufficient to generalize to previously unseen environments. Third, we train

the discriminator using *alignment-based* similarity metric that enables the model to align same concepts in the language and visual modalities. We provide a qualitative assessment of the alignment learned by the model. Finally, we show that a navigation agent, when initialized with components of fully-trained discriminator, outperforms the existing benchmark on success rate by over 10% relative measure on previously unseen environments.

2 The Room-to-Room Dataset

Room-to-Room (R2R) is a visually-grounded natural language navigation dataset in photo-realistic environments (Anderson et al., 2018b). Each environment is defined by a graph where nodes are locations with egocentric panoramic images and edges define valid connections for agent navigation. The navigation dataset consists of language instructions paired with reference paths, where each path is defined by a sequence of graph nodes. The data collection process is based on sampling pairs of start/end nodes and defining the shortest path between the two. Furthermore the collection process ensures no paths are shorter than 5m and must be between 4 to 6 edges. Each sampled path is associated with 3 natural language instructions collected from Amazon Mechanical Turk with an average length of 29 tokens from a vocabulary of 3.1k tokens. Apart from the training set, the dataset includes two validation sets and a test set. One of the validation sets includes new instructions on environments overlapping with the training set (Validation Seen), and the other is entirely disjoint from the training set (Validation Unseen).

Several metrics are commonly used to evaluate agents’ ability to follow navigation instructions. *Path Length (PL)* measures the total length of the predicted path, where the optimal value is the length of the reference path. *Navigation Error (NE)* measures the distance between the last nodes in the predicted path and the reference path. *Success Rate (SR)* measures how often the last node in the predicted path is within some threshold distance d_{th} of the last node in the reference path. More recently, Anderson et al. (2018a) proposed the *Success weighted by Path Length (SPL)* measure that also considers whether the success criteria was met (i.e., whether the last node in the predicted path is within some threshold d_{th} of the reference path) and the normalized path length. Agents should minimize NE and maximize SR and SPL.

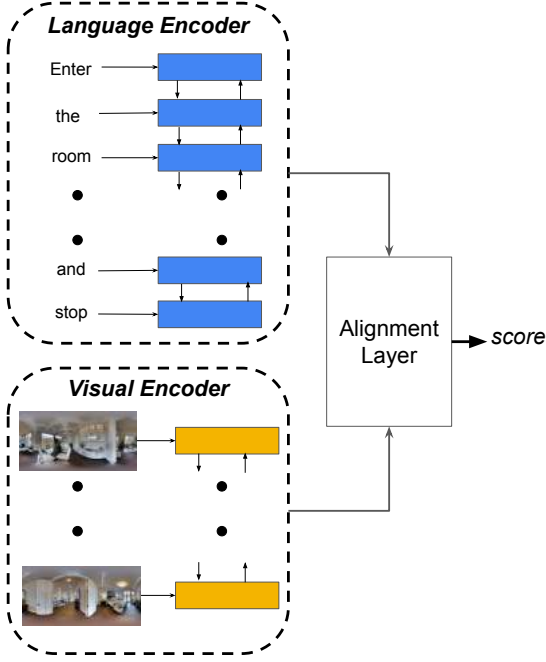


Figure 1: Overview of the discriminator model structure. Alignment layer corresponds to Eq.5,6,7

3 Discriminator Model

VLN tasks are composed of instruction-path pairs, where a path is a sequence of connected locations along with their corresponding perceptual contexts in some environment. While the core task is to create agents that can follow the navigation instructions to reproduce estimates of reference paths, we instead explore models that focus on the simpler problem of judging whether an instruction-path pair are a good match for one another. These models would be useful in measuring the quality of machine-generated instruction-path pairs. Another reasonable expectation from such models would be that they are also able to align similar concepts in the two modalities, e.g., in an instruction like “*Turn right and move forward around the bed, enter the bathroom and wait there.*”, it is expected that the word *bed* is better aligned with a location on the path that has a bed in the agent’s egocentric view.

To this effect, we train a discriminator model that learns to delineate positive instruction-path pairs from negative pairs sampled using different strategies described in Sec.3.2. The discrimination is based on an alignment-based similarity score that determines how well the two input sequences align. This encourages the model to map perceptual and textual signals for final discrimination.

3.1 Model Structure

We use a two-tower architecture to independently encode the two sequences, with one tower encoding the token sequence x_1, x_2, \dots, x_n in the instruction \mathcal{X} and another tower encoding the visual input sequence v_1, v_2, \dots, v_m from the path \mathcal{V} . Each tower is a bi-directional LSTM (Schuster and Paliwal, 1997) which constructs the latent space representation H of a sequence i_1, i_2, \dots, i_k following:

$$H = [h_1; h_2; \dots; h_k] \quad (1)$$

$$h_t = g(\vec{h}_t, \overleftarrow{h}_t) \quad (2)$$

$$\vec{h}_t = LSTM(i_t, \vec{h}_{t-1}) \quad (3)$$

$$\overleftarrow{h}_t = LSTM(i_t, \overleftarrow{h}_{t+1}) \quad (4)$$

where g function is used to combine the output of forward and backward LSTM layers. In our implementation, g is the concatenation operator.

We denote the latent space representation of instruction \mathcal{X} as H^X and path \mathcal{V} as H^V and compute the alignment-based similarity score as following:

$$A = H^X (H^V)^T \quad (5)$$

$$\{c\}_{l=1}^{l=X} = \text{softmax}(A^l) \cdot A^l \quad (6)$$

$$\text{score} = \text{softmax}(\{c\}_{l=1}^{l=X}) \cdot \{c\}_{l=1}^{l=X} \quad (7)$$

where $(.)^T$ is matrix transpose transformation, A is the alignment matrix whose dimensions are $[n, m]$, A^l is the l -th row vector in A and $\text{softmax}(Z) = \frac{\exp^{-Z_j}}{\sum \exp^{-Z_j}}$. Eq.6 corresponds to taking a softmax along the columns and summing the columns, which amounts to content-based pooling across columns. Then we apply softmax operation along the rows and sum the rows up to get a scalar in Eq.7. Intuitively, optimizing this score encourages the learning algorithm to construct the best worst-case sequence alignment between the two input sequences in latent space.

3.2 Training

Training data consists of instruction-path pairs which may be similar (positives) or dissimilar (negatives). The training objective maximizes the log-likelihood of predicting higher alignment-based similarity scores for similar pairs.

We use the human annotated demonstrations in the R2R dataset as our positives and explore three strategies for sampling negatives. For a given

Learning	PS	PR	RW	AUC
no-curriculum	✓			64.5
no-curriculum		✓		60.5
no-curriculum			✓	63.1
no-curriculum	✓	✓		72.1
no-curriculum		✓	✓	66.0
no-curriculum	✓		✓	70.8
no-curriculum	✓	✓	✓	72.0
curriculum	✓	✓	✓	76.2

Table 1: Results on training in different combinations of datasets and evaluating against validation dataset containing PR and RW negatives only.

instruction-path pair, we sample negatives by keeping the same instruction but altering the path sequence by:

- *Path Substitution (PS)* – randomly picking other paths from the same environment as negatives.
- *Partial Reordering (PR)* – keeping the first and last nodes in the path unaltered and shuffling the intermediate locations of the path.
- *Random Walks (RW)* – sampling random paths of the same length as the original path that either (1) start at the same location and end sufficiently far from the original path or (2) end at the same location and start sufficiently far from the original path.

4 Results

Our experiments are conducted using the R2R dataset (Anderson et al., 2018b). Recently, Fried et al. (2018) introduced an augmented dataset (referred to as *Fried-Augmented* from now on) that is generated by using a speaker model and they show that the models trained with both the original data and the machine-generated augmented data improves agent success rates.

We show three main results. First, the discriminator effectively differentiates between high-quality and low-quality paths in *Fried-Augmented*. Second, we rank all instruction-path pairs in *Fried-Augmented* with the discriminator and train with a small fraction judged to be the highest quality—using just the top 1% to 5% (the highest quality pairs) provides most of the benefits derived from the entirety of *Fried-Augmented* when

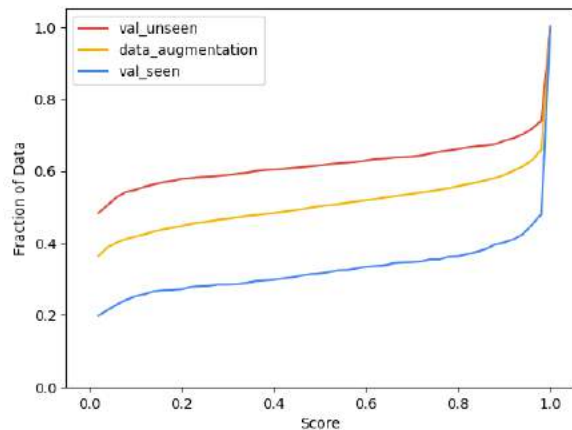


Figure 2: Cumulative distributions of discriminator scores for different datasets. The mean of distribution for R2R *validation seen*, *Fried-Augmented* and R2R *validation unseen* is 0.679, 0.501, and 0.382 respectively.

generalizing to previously unseen environments. Finally, we initialize a navigation agent with the visual and language components of the trained discriminator. This strategy allows the agent to benefit from the discriminator’s multi-modal alignment capability and more effectively learn from the human-annotated instructions. This agent outperforms existing benchmarks on previously unseen environments as a result.

4.1 Discriminator Results

We create two kinds of dataset for each of the negative sampling strategies defined in Section 3.2 – a training set from paths in R2R *train* split and validation set from paths in R2R *validation seen* and *validation unseen* splits. The area-under ROC curve (AUC) is used as the evaluation metric for the discriminator. From preliminary studies, we observed that the discriminator trained on dataset containing PS negatives achieved AUC of 83% on validation a dataset containing PS negatives only, but fails to generalize to validation set containing PR and PW negatives (AUC of 64.5%). This is because it is easy to score PS negatives by just attending to first or last locations, while scoring PR and PW negatives may require carefully aligning the full sequence pair. Therefore, to keep the task challenging, the validation set was limited to contain validation splits from PR and RW negative sampling strategies only. Table 1 shows the results of training the discriminator using various combinations of negative sampling.

Generally, training the discriminator with PS







Dataset	Score	Example
Fried-Augmented	0.001	 <p><i>Walk out of the <u>bedroom</u> and turn left. Walk past the couch and turn right. Walk past the pool and stop on the second step.</i></p>
	0.999	 <p><i>Go up the stairs and turn left. Walk past the kitchen and dining table. Stop behind the dining table.</i></p>
Validation Seen	0.014	 <p><i>Walk across the patio, stop at <u>hanging basket chair</u>.</i></p>
	0.999	 <p><i>Leave the closet and take a right into the hallway. In the hall walk straight and stick left passing a cabinet on your left. Once past the cabinet go into the first room on your left.</i></p>
Validation Unseen	0.00004	 <p><i>Exit the room then turn left and go up the steps then <u>turn right and turn right</u> and wait near the beige couches.</i></p>
	0.9808	 <p><i>Walk down the stairs, at the landing enter the second doorway on the left. Wait near the bookshelf.</i></p>

Table 2: Selected samples from datasets with discriminator scores.

negatives helps model performance across the board. Simple mismatch patterns in PS negatives help bootstrap the model with a good initial policy for further fine-tuning on tougher negatives patterns in PR and RW variations. For example in PS negatives, a path that starts in a bathroom does not match with an instruction that begins with “*Exit the bedroom.*”—this would be an easy discrimination pair. In contrast, learning from just PR and RW negatives fails to reach similar performance. To further confirm this hypothesis, we train a discriminator using curriculum learning (Bengio et al., 2009) where the model is first trained on only PS negatives and then fine-tuned on PR and RW negatives. This strategy outperforms all others, and the resulting best performing discriminator is used for conducting studies in the following subsections.

Discriminator Score Distribution Fig.2 shows the discriminator’s score distribution on different R2R datasets. Since Fried-Augmented contains paths from houses seen during training, it would be expected that discriminator’s scores on *validation seen* and Fried-Augmented datasets be the same if the data quality is comparable. However there is a clear gap in the discriminator’s con-

fidence between the two datasets. This matches our subjective analysis of Fried-Augmented where we observed many paths had clear starting/ending descriptions but the middle sections were often garbled and had little connection to the perceptual path being described. Table 2 contains some samples with corresponding discriminator scores.

Finally we note that the discriminator scores on *validation unseen* are rather conservative even though the model differentiates between positives and negatives from validation set reasonably well (last row in Table 1).

4.2 Training Navigation Agent

We conducted studies on various approaches to incorporate selected samples from Fried-Augmented to train navigation agents and measure their impact on agent navigation performance. The studies illustrate that navigation agents have higher success rates when they are trained on higher-quality data (identified by discriminator) with sufficient diversity (introduced by random sampling). When the agents are trained with mixing selected samples from

Dataset size	Strategy	PL		NE ↓		SR ↑		SPL ↑	
		U	S	U	S	U	S	U	S
1%	Top	11.2	11.1	8.5	8.5	20.4	21.2	16.6	17.6
	Bottom	10.8	10.7	8.9	9.0	15.4	16.3	14.1	13.1
	Random Full	11.7	12.5	8.1	8.3	22.1	21.2	17.9	16.6
	Random Bottom	14.2	15.8	8.4	8.1	19.7	21.7	14.3	15.6
	Random Top	15.9	15.6	7.9	7.6	22.6	25.4	15.2	14.8
2%	Top	11.3	11.7	8.2	7.9	22.3	25.5	18.5	21.0
	Bottom	11.4	14.5	8.4	9.1	17.5	17.7	14.1	12.7
	Random Full	13.3	10.8	7.9	7.9	24.3	25.5	18.2	22.7
	Random Bottom	15.2	18.2	8.1	8.1	20.5	20.8	11.8	16.0
	Random Top	12.9	14.0	7.6	7.5	25.6	25.8	19.5	19.7
5%	Top	17.6	16.9	7.7	7.2	24.6	28.2	14.4	18.2
	Bottom	10.0	10.2	8.3	8.2	20.1	23.2	17.1	19.4
	Random Full	17.8	21.4	7.3	7.0	27.2	29.1	16.4	14.3
	Random Bottom	16.3	10.4	7.9	8.3	22.1	23.0	14.2	20.1
	Random Top	20.0	15.0	7.0	6.9	27.7	30.6	14.8	22.1

Table 3: Results on R2R validation unseen paths (U) and seen paths (S) when trained only with small fraction of Fried-Augmented ordered by discriminator scores. For *Random Full* study, examples are sampled uniformly over entire dataset. For *Random Top/Bottom* study, examples are sampled from top/bottom 40% of ordered dataset. SPL and SR are reported as percentages and NE and PL in meters.

Fried-Augmented to R2R *train* dataset, only the top 1% from Fried-Augmented is needed to match the performance on existing benchmarks.

Training Setup. The training setup of the navigation agent is identical to Fried et al. (2018). The agent learns to map the natural language instruction \mathcal{X} and the initial visual scene v_1 to a sequence of actions $a_{1..T}$. Language instructions $\mathcal{X} = x_{1..n}$ are initialized with pre-trained GloVe word embeddings (Pennington et al., 2014) and encoded using a bidirectional RNN (Schuster and Paliwal, 1997). At each time step t , the agent perceives a 360-degree panoramic view of its surroundings from the current location. The view is discretized into m view angles ($m = 36$ in our implementation, 3 elevations x 12 headings at 30-degree intervals). The image at view angle i , heading angle ϕ and elevation angle θ is represented by a concatenation of the pre-trained CNN image features with the 4-dimensional orientation feature $[\sin \phi; \cos \phi; \sin \theta; \cos \theta]$ to form $v_{t,i}$. As in Fried et al. (2018), the agent is trained using *student forcing* where actions are sampled from the model during training, and supervised using a shortest-path action to reach the goal state.

Training using Fried-Augmented only. The experiments in Table 3 are based on training a navigation agent on different fractions of the

Fried-Augmented dataset ($X = \{1\%, 2\%, 5\%\}$) and sampling from different parts of the discriminator score distribution (*Top*, *Bottom*, *Random Full*, *Random Top*, *Random Bottom*). The trained agents are evaluated on both *validation seen* and *validation unseen* datasets.

Not surprisingly, the agents trained on examples sampled from the *Top* score distribution consistently outperform the agents trained on examples from the *Bottom* score distribution. Interestingly, the agents trained using the *Random Full* samples is slightly better than agents trained using just the *Top* samples. This suggests that the agent benefits from higher diversity samples. This is confirmed by the study *Random Top* where the agents trained using high quality samples with sufficient diversity consistently outperform all other agents.

Training using both R2R *train* and Fried-Augmented. To further investigate the utility of the discriminator, the navigation agent is trained with the full R2R *train* dataset (which contains human annotated data) as well as selected fractions of Fried-Augmented¹. Table 4 shows the results.

¹We tried training on Fried-Augmented first and then fine-tuning on R2R *train* dataset, as done in Fried et al. (2018), but didn't find any appreciable difference in agent's performance in any of the experiments.

Dataset	PL		NE ↓		SR ↑		SPL ↑	
	U	S	U	S	U	S	U	S
Benchmark ²	-	-	6.6	3.36	35.5	66.4	-	-
0%	17.8	18.5	6.8	5.3	32.1	46.1	21.9	30.3
1%	12.5	11.2	6.4	5.7	35.2	45.3	28.9	39.1
2%	14.5	15.1	6.5	5.5	35.7	44.6	27.0	34.1
5%	17.0	12.9	6.1	5.6	36.0	44.8	23.6	37.0
40%	14.9	11.9	6.4	5.5	36.5	49.1	27.1	43.4
60%	16.8	15.7	6.3	5.3	36.0	47.2	24.7	35.4
80%	17.1	18.5	6.2	5.2	35.8	45.0	23.8	29.6
100%	15.6	15.9	6.4	4.9	36.0	51.9	29.0	43.0

Table 4: Results³ on R2R validation unseen (U) and validation seen (S) paths when trained with full training set and selected fraction of `Fried-Augmented`. SPL and SR are reported as percentages and NE and PL in meters.

Method	Split	PL	NE ↓	SR ↑	SPL ↑
Speaker-Follower model (Fried et al., 2018)	U	-	6.6	35.5	-
	S	-	3.36	66.4	-
Speaker-Follower model (our implementation)	U	15.6	6.4	36.0	29.0
	S	15.9	4.9	51.9	43.0
Our implementation, using discriminator pre-training	U	16.7	5.9	39.1	26.8
	S	15.4	5.0	50.4	39.1

Table 5: Results on R2R validation unseen (U) and validation seen (S) paths after initializing navigation agent’s instruction and visual encoders with discriminator.

Validation Unseen: The performance of the agents trained with just 1% `Fried-Augmented` matches with benchmark for NE and SR. With just 5% `Fried-Augmented`, the agent starts outperforming the benchmark for NE and SR. Since `Fried-Augmented` was generated by a speaker model that was trained on R2R *train*, the language diversity in the dataset is limited, as evidenced by the unique token count: R2R *train* has 2,602 unique tokens while `Fried-Augmented` has only unique 369 tokens. The studies show that only a small fraction of top scored `Fried-Augmented` is needed to augment R2R *train* to achieve the full performance gain over the benchmark.

Validation Seen: Since `Fried-Augmented` contains paths from houses seen during training, mixing more of it with R2R *train* helps the agent overfit on *validation seen*. Indeed, the model’s performance increases nearly monotonically on NE and SR as higher fraction of `Fried-Augmented` is mixed in the training data. The agent performs best when it is trained on all of

`Fried-Augmented`.

Initializing with Discriminator. To further demonstrate the usefulness of the discriminator strategy, we initialize a navigation agent’s instruction and visual encoder using the discriminator’s instruction and visual encoder respectively. We note here that since the navigation agent encodes the visual input sequence using LSTM, we re-train the best performing discriminator model using LSTM (instead of bidirectional-LSTM) visual encoder so that the learned representations can be transferred correctly without any loss of information. We observed a minor degradation in the performance of the modified discriminator. The navigation agent so initialized is then trained as usual using *student*

²For a fair comparison, the benchmark is the Speaker-Follower model from Fried et al. (2018) which uses panoramic action space and augmented data, but no beam search (pragmatic inference).

³Our results of the agents trained on the full R2R *train* and 100% `Fried-Augmented` match with Speaker-Follower benchmark on validation unseen but are lower on validation seen. This is likely due to differences in model capacity, hyper-parameter choices and image features used in our implementation. The image features used in our implementation are obtained through a convolutional network trained with a semantic ranking objective on a proprietary image dataset with over 100+ million images (Wang et al., 2014).



Figure 3: Alignment matrix (Eq.5) for discriminator model trained (a) with curriculum learning on the dataset containing PS, PR, RW negatives (b) without curriculum learning on the dataset with PS negatives only. Note that darker means higher alignment.

forcing. The agent benefits from the multi-modal alignment learned by the discriminator and outperforms the benchmark on the *Validation Unseen* set, as shown in Table 5. This is the condition that best informs how well the agent generalizes. Nevertheless, performance drops on *Validation Seen*, so further experimentation will hopefully lead to improvements on both.

4.3 Visualizing Discriminator Alignment

We plot the alignment matrix A (Eq.5) from the discriminator for a given instruction-path pair to try to better understand how well the model learns to align the two modalities as hypothesized. As a comparison point, we also plot the alignment matrix for a model trained on the dataset with PS negatives only. As discussed before, it is expected that the discriminator trained on the dataset containing only PS negatives tends to exploit easy-to-find patterns in negatives and make predictions without carefully attending to full instruction-path sequence.

Fig.3 shows the difference between multi-modal

alignment for the two models. While there is no clear alignment between the two sequences for the model trained with PS negatives only (except maybe towards the end of sequences, as expected), there is a visible diagonal pattern in the alignment for the best discriminator. In fact, there is appreciable alignment at the correct positions in the two sequences, e.g., the phrase *exit the door* aligns with the image(s) in the path containing the object *door*, and similarly for the phrase *enter the bedroom*.

5 Related Work

The release of Room-to-Room (R2R for short) dataset (Anderson et al., 2018b) has sparked research interest in multi-modal understanding. The dataset presents a unique challenge as it not only substitutes virtual environments (e.g., MacMahon et al. (2006)) with photo-realistic environments but also describes the paths in the environment using human-annotated instructions (as opposed to formulaic instructions provided by mapping applications e.g., Cirik et al. (2018)). A number of

methods (Anderson et al., 2018b; Fried et al., 2018; Wang et al., 2018a; Ma et al., 2019a; Wang et al., 2018b; Ma et al., 2019b) have been proposed recently to solve the navigation task described in R2R dataset. All these methods build models for agents that learn to navigate in R2R environment and are trained on the entire R2R dataset as well as the augmented dataset introduced by Fried et al. (2018) which is generated by a speaker model trained on human-annotated instructions.

Our work is inspired by the idea of Generative Adversarial Nets (Goodfellow et al., 2014), which use a discriminative model to discriminate real and fake distribution from generative model. We propose models that learn to discriminate between high-quality instruction-path pairs from lower quality pairs. This discriminative task becomes important for VLN challenges as the data is usually limited in such domains and data augmentation is a common trick used to overcome the shortage of available human-annotated instruction-path pairs. While all experiments in this work focus on R2R dataset, same ideas can easily be extended to improve navigation agents for other datasets like Touchdown (Chen et al., 2018).

6 Conclusion

We show that the discriminator model is capable of differentiating high-quality examples from low-quality ones in machine-generated augmentation to VLN datasets. The discriminator when trained with *alignment based* similarity score on cheaply mined negative paths learns to align similar concepts in the two modalities. The navigation agent when initialized with the discriminator generalizes to instruction-path pairs from previously unseen environments and outperforms the benchmark.

For future work, the discriminator can be used in conjunction with generative models producing extensions to human-labeled data, where it can filter out low-quality augmented data during generation as well as act as a reward signal to incentivize generative model to generate higher quality data. The multi-modal alignment learned by the discriminator can be used to segment the instruction-path pair into several shorter instruction-path pairs which can then be used for creating a curriculum of easy to hard tasks for the navigation agent to learn on. It is worth noting that the trained discriminator model is general enough to be useful for any downstream task which can benefit from such multi-modal align-

ment measure and not limited to VLN task that we use in this work.

References

- Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. 2018a. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.
- Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Stünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. **Curriculum learning**. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 41–48, New York, NY, USA. ACM.
- Howard Chen, Alane Suhr, Dipendra Kumar Misra, Noah Snaveley, and Yoav Artzi. 2018. Touchdown: Natural language navigation and spatial reasoning in visual street environments. *CoRR*, abs/1811.12354.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. **Microsoft COCO captions: Data collection and evaluation server**. *CoRR*, abs/1504.00325.
- Volkan Cirik, Yuan Zhang, and Jason Baldridge. 2018. Following formulaic map instructions in a street simulation environment. In *2018 NeurIPS Workshop on Visually Grounded Interaction and Language*.
- Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In *Neural Information Processing Systems (NeurIPS)*.
- Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. **Generative adversarial nets**. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

- David Harwath, Adrià Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James R. Glass. 2018. [Jointly discovering visual objects and spoken words from raw sensory input](#). In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, pages 659–677.
- Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, and Raia Hadsell Keith Anderson. 2019. [Learning to follow directions in street view](#). *CoRR*, abs/1903.00401.
- Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan Al-Regib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019a. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. 2019b. [The regretful agent: Heuristic-aided navigation through progress estimation](#).
- Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: Connecting language, knowledge, action in route instructions. In *In Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, pages 1475–1482.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.
- Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45:2673–2681.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2017. [Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge](#). *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):652–663.
- Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. [Talk the walk: Navigating new york city through grounded dialogue](#). *CoRR*, abs/1807.03367.
- Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. [Learning fine-grained image similarity with deep ranking](#). In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14*, pages 1386–1393, Washington, DC, USA. IEEE Computer Society.
- Xin Wang, Qiuyuan Huang, Asli Çelikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2018a. [Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation](#). *CoRR*, abs/1811.10092.
- Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. 2018b. [Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation](#). In *Computer Vision – ECCV 2018*, pages 38–55, Cham. Springer International Publishing.
- Haonan Yu and Jeffrey Mark Siskind. 2013. [Grounded language learning from video described with sentences](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 53–63.
- Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 818–833.