Abstract

Vision-and-language navigation requires an agent to navigate through a real 3D environment following a given natural language instruction. Despite significant advances, few previous works are able to fully utilize the strong correspondence between the visual and textual sequences. Meanwhile, due to the lack of intermediate supervision, the agent’s performance at following each part of the instruction remains untrackable during navigation. In this work, we focus on the granularity of the visual and language sequences as well as the trackability of agents through the completion of instruction. We provide agents with fine-grained annotations during training and find that they are able to follow the instruction better and have a higher chance of reaching the target at test time. We enrich the previous dataset with sub-instructions and their corresponding paths. To make use of this data, we propose an effective sub-instruction attention and shifting modules that attend and select a single sub-instruction at each time-step. We implement our sub-instruction modules in four state-of-the-art agents, compare with their baseline model, and show that our proposed method improves the performance of all four agents.

1 Introduction

Creating an agent that can navigate through an unknown environment following natural language instructions has been a dream of human-beings for many years. Such an agent needs to possess the abilities to perceive its environment, understand the instructions and learn the relationship between these two streams of information. Recently, Anderson et al. (2018b) proposed a task called vision-and-language navigation (VLN) that formalized such requirements through an evaluation of an agents ability to follow natural language instructions in photorealistic environments.

Despite the significant progress made by recent approaches, there is little evidence that agents learn the correspondence between observations and instructions. Hu et al. (2019) found that a modified self-monitoring agent (Ma et al., 2019b), could achieve similar performance with (success rate 40.5%) and without (success rate 39.7%) the visual information. Among other reasons such as dataset bias, the result suggests that this agent gains little by having the two streams of information.

We argue that one of the main reasons behind is that the current methods are not adequately teaching agents the relationship between perception—things that the robot is observing—and parts of the instructions. Since datasets do not provide such
information agents can only use the ground-truth trajectory as the teaching signal. Moreover, given the lack of fine-grained annotation, current methods cannot evaluate the (perceptual or linguistic) grounding process at each step as there is no ground truth signal to indicate which part of the instruction has been completed.

To address this problem, we enhance the R2R dataset to acquire intermediate supervision for the agents, providing a fine-grained matching between sub-instructions and the agent’s visual perception, as illustrated in Figure 1, to achieve our Fine-Grained Room-to-Room dataset (FGR2R). We argue that the granularity of the navigation task should be at the level of these sub-instructions, rather than attempting to ground a specific part of the original long and complex instruction without any direct supervision or measure navigation progress at word level.

Our work aims to make the co-grounding process trackable and encourage the agent to navigate precisely on the described path rather than just focusing on the target. We hypothesize that the agent should reach the target with higher success rate by following a detailed instruction with richer information, and in practice, the agent could complete some additional tasks on its way to the target.

In light of this, we propose novel sub-instruction attention to better learn the correspondence between the visual features and the language features. Our agent first segments the long and complicated instruction into short and easy-to-understand sub-instructions using a heuristic based on the grammatical relations provided by Stanford NLP Parser (Qi et al., 2018). Moreover, we propose a shifting module that infers whether the current sub-instruction has been completed. Hence, only one sub-instruction is available to the agent at each time step for textual grounding. These modules can be easily applied to previous agents.

We conduct experiments comparing four state-of-the-art agents to evaluate with and without our sub-instruction module, for agents based on imitation learning (Anderson et al., 2018b; Fried et al., 2018; Ma et al., 2019b) and reinforcement learning (Wang et al., 2019). These results show the effectiveness of the intermediate supervision and our proposed modules. Furthermore, we demonstrate the trackability of the navigation process through qualitative and quantitative analysis.

2 Related Work

Visual and textual grounding. Visual grounding aims to learn the relationship between a natural-language description and a spatial or temporal region in an image or video, respectively. It is an essential component for a variety of tasks in computer vision such as visual question answering (VQA) (Schwartz et al., 2017; Anderson et al., 2018a; Yu et al., 2019; Hudson and Manning, 2019), image captioning (Xu et al., 2015; Karpathy and Fei-Fei, 2015; Anderson et al., 2018a; Cornia et al., 2019; Ma et al., 2019a), video understanding (Gao et al., 2017; Anne Hendricks et al., 2017; Ma et al., 2018; Rodriguez et al., 2020), phrase localization (Xiao et al., 2017; Engilberge et al., 2018; Yu et al., 2018), among others. In the case of vision-and-language navigation, agents use grounding to select an image in a panoramic view, defining where the agent will go in the following step (Fried et al., 2018; Ma et al., 2019b; Wang et al., 2019; Tan et al., 2019; Wang et al., 2018; Hu et al., 2019; Landi et al., 2019; Ma et al., 2019c; Ke et al., 2019).

Vision and language navigation. Anderson et al. (2018b) formalized the vision-and-language navigation task in photo-realistic environment, and proposed a benchmark Room-to-Room dataset and a sequence-to-sequence agent as a baseline model. Other datasets in real environment, such as R4R (Jain et al., 2019) which is an extended version of R2R with longer instruction-path pairs, and Touchdown (Chen et al., 2019) which is in street views have also been proposed for study.

Researchers have addressed the task through a great variety of approaches. Wang et al. (2018) propose a look-ahead model that combines model-based and model-free reinforcement learning, predicts the next agent’s state and reward during navigation. Fried et al. (2018) proposed the Speaker-Follower model which generates augmented samples for training and makes use of the panoramic action space to ground and navigate efficiently. Later, Ma et al. (2019b) introduced the Self-Monitoring agent which includes a vision and language co-grounding network and a progress monitor. The progress monitor estimates a normalized distance to the target and guides the transition of the textual attention. Wang et al. (2019) applied the REINFORCE algorithm (Williams, 1992) to improve the agent’s generalizability and proposed a Self-Supervised Imitation Learning (SIL) method to facilitate lifelong learning in a new environment.
Very recently, Zhu et al. (2019) and Hao et al. (2020) pretrain the network. The VLN task requires the agent to navigate through a real environment to a target location following a natural language instruction.

Formally, an instruction $w$ is a sequence of words $\langle w_1, w_2, \ldots, w_l \rangle$ provided to the agent at the beginning of its navigation, where $w_i$ denotes the $i$-th word in the sequence. The environment is defined as set of viewpoints $\{p_j\}$ denoting all the navigable locations. At time step $t$, the agent at viewpoint $p_t$ receives a panoramic view $V_t$ composed of $n$ single view images $\langle v_{t,1}, v_{t,2}, \ldots, v_{t,n} \rangle$. Using the given instruction $w$ and the current observation $V_t$, the agent needs to infer a series of action $\langle a_1, a_2, \ldots, a_n \rangle$, each action triggers a transition signal from $p_t$ to $p_{t+1}$. The episode ends when the agent output a $STOP$ action or the maximum number of steps allowed is reached.

3 Sub-instruction Aware VLN

The VLN task requires the agent to navigate through a real environment to a target location following a natural language instruction.

In contrast to all previously mentioned methods that ground the complete instruction, we propose to divide the instruction into meaningful semantic sub-instructions, and teach the agent to complete one at a time before reading the next sub-instruction. In that spirit, our method is similar to the image captioning work by Cornia et al. (2019). They design a shifting gate over the image regions to control the visual features that feed in each time step the caption module. We differ from their work in the modality that is attended. Our method works in the language representation, and the shifting depends only on local context rather than looking over all the sub-instructions.

3.1 Base Agent Model

We build our sub-instruction module based on the current state-of-the-art VLN agents, as shown in Figure 2. Those agents share a similar pipeline, a sequence-to-sequence architecture using a recurrent neural network with vision and language attention. In this section, we refer to the Self-Monitoring Agent (Ma et al., 2019b) to present the flow of information in the network.

Visual and textual encoding. Before the start of navigation, the agent first encodes the given instruction, using an LSTM with a learned embedding as $\tilde{w}_j = \text{Embed}(w_j)$ and $u_1, u_2, \ldots, u_l = \text{LSTM}(\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_l)$, where $u_j$ is the hidden state of word $w_j$ in the instruction. In the case of the panoramic view, the agent encodes the images using a pre-trained ResNet-152 (He et al., 2016) on ImageNet (Russakovsky et al., 2015) for each navigable direction. A 4-dimensional vector $[\sin \psi, \cos \psi, \sin \theta, \cos \theta]$ is concatenated with the image encoding to represent the direction of visual features, where $\psi$ and $\theta$ are the heading and elevation angles respectively.

Policy module and co-grounding. We define the agent’s state at time $t$ as a combination of the attended textual representation $\tilde{u}_t$, the attended visual representation $\tilde{v}_t$ and the previous selected action $a_{t-1}$, encoded by an LSTM as

$$h_t, m_t = \text{LSTM}(\langle \tilde{u}_t; a_{t-1} \rangle, (\tilde{v}_t, m_{t-1})).$$

We refer to $h$ and $m$ as the agent’s state and memory, respectively.

The attended textual representation is obtained by performing soft-attention over the language features $U$ with the agent’s state at the previous time.
step. The attention weights over all the words are calculated as \( z_{t,i}^{\text{text}} = (W_{a}[\hat{h}_t, \hat{u}_t])^T u_j \) and \( \alpha_t = \text{softmax}(z_{t,i}^{\text{text}}) \), obtaining the attended textual representation by \( \hat{u}_t = \alpha_t^T U \). Similarly, we perform soft-attention over the single-view visual features \( V_t \) as \( z_{t,i}^{\text{vis}} = (W_{v} h_{t-1})^T g(v_{t,i}) \) where \( g(\cdot) \) is a multi-layer perceptron (MLP), and the attention weight \( \beta_t = \text{softmax}(z_{t,i}^{\text{vis}}) \). The attended visual representation is \( \hat{v}_t = \beta_t^T V_t \). The previous selected action \( \alpha_{t-1} \) is represented by the visual features at the previously selected action direction.

Finally, the agent decides an action by finding the visual features at a navigable direction with the highest correspondence to the attended language features \( \hat{u} \) and the agent’s current state \( h_t \). The probability at each navigable direction is computed as:

\[
o_{t,i} = (W_{a}[\hat{h}_t, \hat{u}_t])^T g(v_{t,i})
\]

and

\[
p_t = \text{softmax}(o_t)
\]

where \( g(\cdot) \) is the same MLP as in visual attention for feature projection. Then, the agent moves in a panoramic action space (Fried et al., 2018), so that it jumps directly to an adjacent viewpoint at the selected direction.

All baseline agents in our experiments are variants of this pipeline. For instance, the Speaker-Follower agent (Fried et al., 2018) encodes the agent’s state with only the previous action and the attended visual features. In the case of the Back-Translation agent (Tan et al., 2019), it attends the language features by the agent’s current state.

### 3.2 Chunking

To encourage the learning of vision and language correspondences, we provide short and easy-to-learn sub-instructions to the agent at each time step. Formally, for each instruction \( w \), there exists a set of sub-instructions \( X = \{x_1, x_2, \ldots, x_L\} \), where \( x_i = \langle w_j \rangle \) and \( L \) is the total number of sub-instructions. The sub-instructions are mutually exclusive and cover the entire \( w \).

In order to break the original instruction into several sub-instructions, we propose a chunking function, where each sub-instruction is an independent navigation task and usually requires the agent to perform one or two actions to complete.

To achieve this automatically, we design chunking rules based on the grammatical relations between words in the instruction, where the relations are produced by the Stanford NLP Parser (Qi et al., 2018), a pre-trained natural language analysis tool. We first pass the entire instruction into the StanfordNLP Parser for extracting the dependency and the governor of each word, denoted as \( \eta(w_j) \) and \( \rho(w_j) \), respectively. Using the two attributes, we formulate an heuristic following the rules shown in Algorithm 1.

The chunking function considers words in the instruction that meet one of the following three conditions as the beginning of a new sub-instruction: (1) its dependency is \( \text{root} \) and all the words before belong to the previous chunk, (2) its dependency is \( \text{conj} \) and its governor is the previous \( \text{root} \), (3) its dependency is \( \text{parataxis} \) and all the words before belong to the previous chunk.

If any one of the three conditions is met, a \( \text{Check}(\cdot) \) function will be performed on the temporary chunk to decide whether to save the temporary chunk into the final sub-instruction list \( l_X \). Here, the \( \text{Check}(\cdot) \) function examines if the temporary chunk meets two conditions: (1) the chunk length should exceed the minimum length of 2 words, and (2) the temporary chunk only contains a single motion instruction which is following the previous chunk or is leading the next chunk, if yes, then the temporary chunk should be appended to the previous chunk or added to the next chunk respectively. This condition is to prevent sub-instructions such as “go straight then”.

### Algorithm 1 Chunking Function

1: Initialize empty lists \( l_{\text{conj}}, l_x, l_\eta \). Count \( k = 0 \).
2: # Find index of the word that satisfies condition (2)
3: for \( w_j \) in \( w \) do
4: if \( \eta(w_j) = \text{conj} \&\& \rho(w_j) = 1 \) then
5: Save word index \( j \) into \( l_{\text{conj}} \)
6: end if
7: end for
8: for \( w_j \) in \( w \) do
9: # Condition (1)
10: if \( \eta(w_j) = \text{root} \&\& (\text{root} \text{ in } l_\eta \text{ or \text{parataxis} in } l_\eta) \) then
11: \( l_X \leftarrow \text{Check}(l_X) \)
12: # Condition (2)
13: else if \( k \leq \text{len}(l_{\text{conj}}) - 1 \&\& \rho(w_j) = l_{\text{conj}}[k] \) then
14: \( l_X \leftarrow \text{Check}(l_X), k = k + 1 \)
15: # Condition (3)
16: else if \( \eta(w_j) = \text{parataxis} \&\& (\text{root} \text{ in } l_\eta \text{ or \text{parataxis} in } l_\eta) \) then
17: \( l_X \leftarrow \text{Check}(l_X) \)
18: # Save the word into temporary chunk
19: else if \( \eta(w_j) = \text{not punct} \) then
20: Save \( w_j \) into \( l_x \), save \( \eta(w_j) \) into \( l_\eta \)
21: end if
22: end for

We provide an illustrative example here. Our chunking function breaks the given instruction “En-
ter through the glass door. Go up the wooden plank stairs on the right. Enter the doorway next to the bear head and wait there.” into 1 “Enter through the glass door”, 2 “Go up the wooden plank stairs on the right”, 3 “Enter the doorway next to the bear head” and 4 “And wait there”, as shown in Figure 1. In the third and the fourth sub-instructions, the words “Enter” and “Wait” satisfy the conditions (1) and (2), respectively. Notice that the governor of conjunction word “And” is “Wait”, so it has been assigned to the fourth sub-instruction.

3.3 Sub-Instruction Module

To encourage the agent to learn the correspondence between visual and language features in a sub-instruction, we modify the base agents to include a sub-instruction module, which enables the agent to focus on a particular sub-instruction at each time step, as shown in Figure 2. It contains two main components: the sub-instruction attention and the sub-instruction shifting module.

Sub-instruction attention. The module attends the words inside the selected sub-instruction $x_i$ through a soft-attention mechanism. Formally, at each time step, we calculate the distribution of weights over each word in the as:

$$z_t^{\text{text}} = (W_u h_{t-1})^T x_{i,j},$$

$$a_t = \text{softmax}(z_t^{\text{text}})$$

where $h_{t-1}$ is the previous state of the agent and $W_u$ is the learned weights. The grounded representation of the sub-instruction is hence $\hat{x}_t = a_t^T x_t$.

With the sub-instruction attention, the agent is forced to attend the most relevant part of the instruction and prevent the agent from “getting distracted” by the other part of the instruction that has been completed or to be completed in the further steps.

Sub-instruction shifting. At each time step, the agent needs to decide whether the current sub-instruction has been completed or not. We enable this functionality by designing a shifting module which estimates the probability of proceeding to the next sub-instruction.

The module uses a recurrent neural architecture to encode a representation that reflect the vision and language co-grounded features:

$$h_t^c = \sigma(W_{c1}[W_{c0}(h_t), v_t^c, \hat{x}_t]) \odot \tanh(m_t)$$  \hspace{1cm} (5)

where $h_t$ and $m_t$ is the agent’s current state and memory, $v_t^c$ is the visual feature at the selected action direction, $\sigma$ represents a sigmoid function, $W_{c1}$ and $W_{c0}$ are the learned weights and $\odot$ denotes the Hadamard product.

Then, the module computes the shifting probability from $h_t^c$ and a one-hot encoding $e_t$ of the number of sub-instructions left to be completed, as:

$$p_t^s = \sigma(W_{c2}(e_t), h_t^c)$$ \hspace{1cm} (6)

where $W_{c2}$ and $W_{c3}$ are the learned parameters. Here, $e_t$ introduces a learnable prior on when to shift before viewing the scene. This prior is then modified by taking into account the visual evidence, which is essential for efficient navigation. If the shifting probability exceed a certain threshold, a shift signal $s_t \in \{0, 1\}$ reading sub-instruction will be produced.

3.4 Training

In the training stage, for each instruction $w$, there exists a corresponding ground-truth path $P_g = \langle p_{g(1)}, p_{g(2)}, \ldots, p_{g(M)} \rangle$. In the case of sub-instructions, we partition the path into sub-paths, one for each sub-instruction.

The binary cross-entropy loss compares the estimated shifting probabilities $p_t^s$ to the target shifting signals $y_t^s$, where the target is either 1 or 0 depending on the distance between the agent’s current position and the ending viewpoint of the current sub-path. In summary, the agent’s parameters are learned to optimized:

$$\mathcal{L} = - \sum_t y_t^s \log p_t^s - \sum_t y_t^\neg s \log (1 - p_t^s)$$ \hspace{1cm} (7)

where $p_t^s$ is the predicted action, $y_t^s$ and $y_t^\neg s$ are the ground-truth action and shifting signal respectively at time step $t$.

During training, we apply student-forcing supervision to the action to encourage exploration, but use teacher-forcing for the sub-instruction shifting (Williams and Zipser, 1989; Anderson et al., 2018b). In early stages of training, the ground-truth shifting signal will have a large number of zeros since the agent has a high probability of deviating from the desired path. We prevent the sub-instruction shifting module from converging to an undesirable local minimum by forcing the shifting
loss to consider an equal number of randomly selected shift and do-not-shift samples in each time step.

4 The FGR2R Dataset

To acquire the matching between vision and language sub-sequences, we introduce a Fine-Grained Room-to-Room Dataset (FGR2R) dataset which enriches the benchmark Room-to-Room dataset by dividing the instructions into sub-instructions and pairing each of those with their corresponding viewpoints in the path.

4.1 Dataset statistics

FGR2R enriches the R2R dataset which contains instruction-path pairs especially collected for the vision-and-language navigation task (Anderson et al., 2018b). R2R possesses 21,567 navigation instructions and 7,189 paths in 91 real-world environments, where 3 or 4 different natural language instructions describe each path. The R2R data has been split for learning proposes, with 4,675 paths for training and 340 paths for seen validation in 61 scenes, 783 paths in 11 scenes for unseen validation and the remaining 1,391 paths in 18 scenes for testing. Based on the original R2R data, FGR2R divides the instructions for the training and validation set in an average of 3.1 sub-instructions. Each sub-instruction has 8.6 words on average. Sub-instructions are paired on average 2.6 viewpoints, and with a minimum and maximum of 2 and 7 viewpoints respectively, as can be seen in Figure 1.

4.2 Dataset collection

We add annotations to the accessible training and validation set of the data using Amazon Mechanical Turk (AMT). We generate the sub-instructions automatically from the original R2R dataset using the chunking function, introduced in Section 3.2. To evaluate the quality of the generated sub-instructions, we compare the output sub-instructions against a manually annotated subset of 300 samples, obtaining a smooth BLEU-4 score of 0.84. We build a web interface which shows an interactive window of the environment, the original instruction, the sub-instructions, and a drop-down list besides each sub-instruction for assigning the corresponding start and end viewpoints. The annotator can freely move on the ground-truth path and freely rotate the camera to observe its surroundings. We ask annotators to partition the ground truth path and assign a sub-instruction to those partitions. To ensure the quality, we exam each annotator with 15 ground-truth samples. In total, there are 126 participants, we reject workers with low agreement to the ground-truth. This leaves 58 annotators qualified to complete the annotation task.

5 Experiments

5.1 Experiment Setup

We experiment with four state-of-the-art VLN agents with and without our sub-instruction module and compare their performance on the original R2R unseen validation set.

The agents are chosen to include the most common network architectures, training strategies and inference methods among the previous VLN agents. They include the Sequence-to-Sequence (Seq2Seq) (Anderson et al., 2018b) model which does not apply panoramic action space, two vision-and-language co-grounding models, the Speaker-Follower (Fried et al., 2018) and the Self-Monitoring agent (Ma et al., 2019b), as well as the Back-Translation model (Tan et al., 2019) which applies reinforcement learning. For all agents, we implement our sub-instruction module in their network based on their officially released code. For the self-monitoring agent, we remove the progress monitor since it requires the attention weight over the entire instruction for estimating the navigation progress.

Implementation Details. To obtain the word representations in each sub-instruction, the entire instruction is first passed to a unidirectional LSTM, then we implement chunking on the language hidden states to obtain the word representations of the selected sub-instructions.

The ground-truth shifting signal at each time-step is dependent on the distance between the agent’s current position and the end viewpoint of the selected sub-instruction. If the distance is smaller than or equal to 0.5 meters, the ground-truth shift signal \( s_t \) will be 1, and 0 otherwise.

For the Back-Translation model (Tan et al., 2019), we only apply chunk shifting loss to the teacher-forcing imitation learning branch, so that the agent walks on the ground-truth path and learns the chunk-shifting with less noise.

Evaluation Metrics. We follow the same metrics that previous work employed for evaluating

---

3 More information about R2R can be found in Matterport3D dataset (Chang et al., 2017) and R2R dataset (Anderson et al., 2018b)
**Instruction:** Take a right and then take a left and walk out of the bathroom. Wait on the carpet in the room to the left.

(a) Self-Monitoring agent without sub-instruction module:

- Error: 2.81m
- nDTW: 0.68
- Stop: by reaching the maximum steps

Sub-instruction 1:

- Take a right.
- \( s_x = 1 \)

Sub-instruction 2:

- And then take a left.
- \( s_y = 1 \)

(b) Self-Monitoring agent with sub-instruction module:

- Error: 0.00m
- nDTW: 1.00
- Stop: by predicting a STOP action

Sub-instruction 1:

- Take a right.
- \( s_x = 1 \)

Sub-instruction 2:

- And then take a left.
- \( s_y = 0 \)

Sub-instruction 3:

- And walk out of the bathroom.
- \( s_x = 1 \)

Sub-instruction 4:

- Wait on the carpet in the room to the left.
- \( s_y = 0 \)

![Figure 3](image_url)

Figure 3: Qualitative comparison of a successful case without and with sub-instruction module. Without sub-instruction module, the agent fails to follow the instruction and stops next to the target by chance. With sub-instruction module, the agent navigates on the described path and eventually stops right at the target location. For panoramic visualization and more examples please refer to the supplementary material.

### Table 1: Comparison on the unseen validation set with (FGR2R) and without (R2R) the sub-instruction module.

| # | Model                       | PL  | NE  | OSR | SR  | SPL | nDTW |
|---|-----------------------------|-----|-----|-----|-----|-----|-----|
| 1 | Seq2Seq (Anderson et al., 2018b) | 8.34 | 7.85 | 29.2 | 22.9 | | |
| 2 | Speaker-Follower (Fried et al., 2018) | 13.57 | 6.66 | 44.8 | 34.7 | | |
| 3 | Self-Monitoring* (Ma et al., 2019b) | 13.95 | 6.16 | 53.7 | 42.4 | | |
| 4 | Back-Translation (Tan et al., 2019) | 9.81 | 5.67 | 54.8 | 46.7 | | |

*Self-Monitoring agent without progress monitor, the reported SR is 42% in the original paper.

6 Results and Analysis

We compare the performance of the four agents on the R2R unseen validation set. We also present the trackability of the navigation process resulting from our FGR2R data.

6.1 Comparisons

**Quantitative results.** Table 1 shows the results of the four agents in unseen environments. The performance of the imitation learning agents (Row 1-3) with our sub-instruction attention module outperforms the base agents. In terms of the success rate, the Seq2Seq, Speaker-Follower and the Self-Monitoring agents achieve an absolute increase of 1.1%, 4.9% and 1.7% respectively. The improvement is consistent in other metrics, e.g. for the Self-Monitoring agent, its SPL improves from 0.30 to 0.32 and its nDTW score grows from 0.58 to 0.61. The overall improvement on Path Length and nDTW score for the first three agents indicates that using sub-instructions improves the agent’s ability to navigate on the described path. As for the Back-Translation agent (Row 4), the performance with sub-instruction attention is very similar to the baseline, one possible reason could be that the introduction of sub-instruction shifting perturbs the learning of action, especially for the reinforcement learning scheme which the agent could deviate far from the ground-truth path.

Learning when the agent needs to read a new sub-instruction is a difficult task, since the shifting signal depends on the language and visual relationship. A viewpoint in a specific environment can be considered as shifting or not depending on the sub-instruction that the agent follows. In Table 2, we compute the confusion matrix of the shifting...
signal for each agent at their visited viewpoints. Given the imbalance in shifting signal, we use the accuracy, precision, recall and F1-score to evaluate the performance of our module. Results show that all the agents have huge room for improvement for shifting, we consider these results to be useful baselines for any future methods that use sub-instructions. Notice that agents visit a different number of viewpoints due to the maximum number of steps allowed, the use of panoramic action space and the ability to stop. In the case of Seq2Seq model, since the agent is not using a panoramic view, it performs many actions to change the camera orientation.

**Qualitative performance.** We illustrate a qualitative example in Figure 3 to show how the sub-instruction module works in the agent. In the example, both the baseline model and the model with the sub-instruction module completes the task successfully. However, unlike the baseline model which fails to follow the instruction and stops within 3 meters of the target by chance, our model correctly identifies the completeness of each sub-instruction, guides the agent to walk on the described path and eventually stops right at the target position.

**Global versus local context.** In an indoor navigation task, both the global and local context are critical. Global context offers high-level guidance for long-term exploration, while local context helps exploitation. For human, global context could be more informative because people have learnt a very good scene awareness of the buildings from their experience. However, for agents which learn to navigate from scratch, asking them to ground the global context to a local scene without any direct supervision could result in an agent which cannot understand and follow detailed instructions but only care about the target. Our sub-instruction module facilitates the agent to learn to ground local context, resulting in an agent which values each sub-instruction and moves towards the target step-by-step.

### Table 2: Confusion matrix, and Precision, Recall and F1-Score of the shifting signal on the unseen validation set.

| #   | Model with sub-instructions | SR | TN   | FN   | FP   | TP   | Accuracy | Precision | Recall | F1-Score |
|-----|----------------------------|----|------|------|------|------|----------|-----------|--------|----------|
| 1   | Seq2Seq (Anderson et al., 2018b) | 22.9 | 21436 | 4796 | 1602 | 608  | 0.775    | 0.275     | 0.113  | 0.160    |
| 2   | Speaker-Follower (Fried et al., 2018) | 34.7 | 3983  | 452  | 4878 | 963  | 0.481    | 0.165     | 0.680  | 0.265    |
| 3   | Self-Monitoring (Ma et al., 2019b) | 42.4 | 2666  | 4532 | 409  | 1284 | 0.444    | 0.758     | 0.221  | 0.342    |
| 4   | Back-Translation (Tan et al., 2019) | 46.7 | 2181  | 4765 | 303  | 1256 | 0.445    | 0.806     | 0.209  | 0.331    |

### Table 3: Performance of sub-instruction clusters ranked by mean distance, $\bar{d}$, $f$ and $\bar{\tau}$ above denote distance, frequency and number of viewpoints respectively.

| rank | $\bar{d}$ | $\bar{\tau}$ | $f$ | Representative sub-instruction |
|------|-----------|--------------|-----|--------------------------------|
| 1    | 2.10      | 0.85         | 12  | exit room                      |
| 2    | 2.35      | 0.68         | 19  | walk up two flight of stairs   |
| 3    | 2.38      | 0.72         | 23  | exit the bedroom               |
| 4    | 2.47      | 0.72         | 104 | walk out of the bedroom        |
| 5    | 2.51      | 0.75         | 198 | go up the stairs               |
| 96   | 6.05      | 0.49         | 74  | with the storage to your right exit |
| 97   | 6.41      | 0.46         | 53  | after descend two flight of stairs |
| 98   | 6.71      | 0.50         | 46  | after you leave the bathroom  |
| 99   | 7.04      | 0.45         | 78  | walk straight when you get to the |
| 100  | 7.56      | 0.36         | 38  | wait by the bathtub            |

**6.2 Trackability**

With the FGR2R data, we reveal the navigation process of the agent working on specific sub-instructions. For each sub-instruction, we measure the similarity between the ground-truth path and the actual trajectory using nDTW score as well as the distance between the end viewpoint of the sub-instruction and the predicted shift viewpoint. As a result, we can estimate the performance of the agent in each sub-task. Moreover, we cluster the sub-instructions into 100 clusters using complete-linkage hierarchical agglomerative clustering algorithm aiming to understand what type of sub-instruction the agent can follow. However, instead of using a standard metric of distance such as Euclidean distance, we compute a similarity matrix of sub-instructions using the BLEU-4 metric. We experiment with the Self-Monitoring agent and present a summary of the top five and the bottom five clusters ranked by the mean distance, as shown in Table 3.

We can see from the table that the clusters on which the agent performs better consist of simple and direct sub-instructions which refer to a single action, such as “exit the bathroom” and “go up the stairs”. On the other hand, with sub-instructions that refer to specific objects or express an action which is conditioned on the completion of another action, such as “with the storage to your right exit …”, the agent deviates far from the described path. Moreover, the ranking does not show a strong
correlation with the frequency or the number of viewpoints of each sub-instruction. These results suggest that agent is incapable of understanding complex natural language instructions or ground to specific objects with a high accuracy.

7 Conclusion

In this paper we introduce a novel sub-instruction module and the Fine-Grained R2R Dataset to encourage the learning of correspondences between vision and language. The sub-instruction module enables the agent to attend to one particular sub-instruction at each time-step and decides whether the agent needs to proceed to the next sub-instruction. Our experiments show that by implementing the sub-instruction module in state-of-the-art agents, those agents are able to follow the given instruction more closely and achieve better performance. We also show that, with the sub-instruction annotations, the entire navigation trajectory is trackable. We believe that the idea of sub-instruction module and a sub-instruction annotated dataset can benefit future studies in the VLN task as well as other vision-and-language problems.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6077–6086.

Peter Anderson, 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, pages 3674–3683.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In The IEEE International Conference on Computer Vision (ICCV).

Angel Chang, Angela Dai, Thomas Funkhouser, Macliej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. In 2017 International Conference on 3D Vision (3DV), pages 667–676. IEEE.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 12538–12547.

Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2019. Show, control and tell: A framework for generating controllable and grounded captions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8307–8316.

Martin Engelberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. 2018. Finding beans in burgers: Deep semantic-visual embedding with localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3984–3993.

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 Advances in Neural Information Processing Systems, pages 3314–3325.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In Proceedings of the IEEE International Conference on Computer Vision, pages 5267–5275.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 2020. Towards learning a generic agent for vision-and-language navigation via pre-training. arXiv preprint arXiv:2002.10638.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.

Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. 2019. Are you looking? grounding to multiple modalities in vision-and-language navigation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6551–6557.

Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhaes, Jason Baldridge, and Eugene Ie. 2019. Transferable representation learning in vision-and-language navigation. In Proceedings of the IEEE International Conference on Computer Vision, pages 7404–7413.

Drew Hudson and Christopher D Manning. 2019. Learning by abstraction: The neural state machine. In Advances in Neural Information Processing Systems, pages 5901–5914.
Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1862–1872.

Andrei Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3128–3137.

Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. 2019. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6741–6749.

Federico Landi, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. 2019. Embodied vision-and-language navigation with dynamic convolutional filters. Proceedings of the British Machine Vision Conference.

Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. 2018. Attend and interact: Higher-order object interactions for video understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6790–6800.

Chih-Yao Ma, Yannis Kalantidis, Ghassan AlRegib, Peter Vajda, Marcus Rohrbach, and Zsolt Kira. 2019a. Learning to generate grounded image captions without localization supervision. arXiv preprint arXiv:1906.00283.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019b. 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. 2019c. The regretful agent: Heuristic-aided navigation through progress estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6732–6740.

Gabriel Magalhaes, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. Effective and general evaluation for instruction conditioned navigation using dynamic time warping. arXiv preprint arXiv:1907.05446.

Volodymyr Mnih, Adria Puigdomech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In International conference on machine learning, pages 1928–1937.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2018. Universal dependency parsing from scratch. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 160–170.

Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. 2020. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In The IEEE Winter Conference on Applications of Computer Vision, pages 2464–2473.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252.

Idan Schwartz, Alexander Schwing, and Tamir Hazan. 2017. High-order attention models for visual question answering. In Advances in Neural Information Processing Systems, pages 3664–3674.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2610–2621.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6629–6638.

Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. 2018. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 37–53.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. Neural computation, 1(2):270–280.

Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. 2017. Weakly-supervised visual grounding of phrases with linguistic structures. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5945–5954.
Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, pages 2048–2057.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1307–1315.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6281–6290.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. 2019. Vision-language navigation with self-supervised auxiliary reasoning tasks. arXiv preprint arXiv:1911.07883.