Action Inference for Destination Prediction in Vision-and-Language Navigation

Anirudh Reddy Kondapally
The University of Tokyo
Honda R&D Co.,Ltd.
anirudh_kondapally-reddy@jp.honda

Kentaro Yamada
Honda R&D Co.,Ltd.
kentaro_yamada@jp.honda

Hitomi Yanaka
The University of Tokyo
hyanaka@is.s.u-tokyo.ac.jp

Abstract

Vision-and-Language Navigation (VLN) encompasses interacting with autonomous vehicles using language and visual input from the perspective of mobility. Most of the previous work in this field focuses on spatial reasoning and the semantic grounding of visual information. However, reasoning based on the actions of pedestrians in the scene is not much considered. In this study, we provide a VLN dataset for destination prediction with action inference to investigate the extent to which current VLN models perform action inference. We introduce a crowd-sourcing process to construct a dataset for this task in two steps: (1) collecting beliefs about the next action for a pedestrian and (2) annotating the destination considering the pedestrian's next action. Our benchmarking results of the models on destination prediction lead us to believe that the models can learn to reason about the effect of the action and the next action on the destination to a certain extent. However, there is still much scope for improvement.

1 Introduction

The widespread belief is that autonomous vehicles and mobility services will become commonplace on the roads. Among methods being investigated as a means for humans to interact with these devices, one of the most intuitive approaches is to use language. Vision-and-Language Navigation (VLN) is a task in which navigation instructions are given in free-form language based on visual information to an autonomous vehicle or mobility. Although there are two broad variations of VLN, i.e. outdoor and indoor, we focus on outdoor VLN to tackle challenges for autonomous vehicles and mobility services. Solving outdoor VLN tasks requires spatial and semantic grounding of the instructions and considerable research has been done for VLN (Chen et al., 2019; Hermann et al., 2020; Vasudevan et al., 2021; Deruyttere et al., 2019).

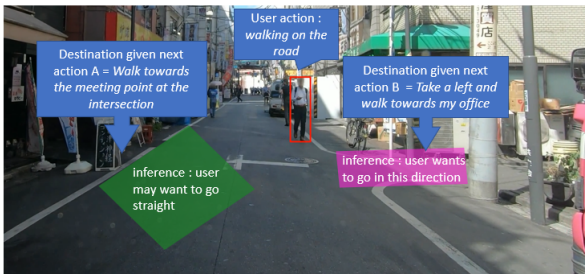

Figure 1: Example illustrating how the green segment is the appropriate destination to pick up the user in case of next action A and the pink segment in case of next action B.

However, none of these works delve into how the actions of pedestrians affect the navigation decisions of the vehicle. We introduce a VLN task of destination prediction for picking up a pedestrian i.e. user of the vehicle in the scene that requires action inference. We define action inference as how the actions and beliefs of the next actions performed by the user in the near future affect the destination to pick up the user. For example in Figure 1, we can see how belief in the next action affects where to pick up the user. To create a dataset annotated with action and next action in a scalable way, we propose annotation processes in two steps shown in Figure 2. In the first step of annotation, we collect the next action knowledge about the likely near future, and in the second step, we collect the destination predictions based on the actions and the next actions.

We test the models used in destination prediction to check whether they can reason about the near future. Our results show that models with insertion of the action and the next action perform better than those without any knowledge insertion. However, since the performance is still quite low, there is room for further improvement.

Our main contributions in this paper are:

- We create a destination prediction dataset for VLN tasks focusing on action inference.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 286–293

- Our experiments using destination prediction models show the importance of action inference and the need to further explore methods to handle action inference.

2 Related Work

StreetNav (Hermann et al., 2020) focuses on using navigation system style street-view hence instructions focus on grounding street names and using spatial cues. In Touchdown (Chen et al., 2019), the instructions help an agent find an object hidden randomly in a street view map, focusing on 3-dimensional instructions along with more variations of words. Talk2Nav (Vasudevan et al., 2021) improves on these datasets by using landmark-based instructions, but it ends up focusing on identifying the landmarks through the use of different ways of referring to them. All of these datasets end up focusing on immovable items such as landmarks or streets instead of taking advantage of the dynamicity of outdoor scenes.

Talk2Car (Deruyttere et al., 2019) has more conversation-style instructions to refer to an object in the scene, but this leads to phrases where the object is mostly present directly in dialogues without any particular consideration given to actions performed in the scene. Talk2Car-RegSeg (Rufus et al., 2021) is a closely associated work to ours and they try to define the destination to navigate to based on the annotation instructions in Talk2Car. However, in contrast to our work, they do not provide any insights into how destinations are affected by action inference. Titan (Malla et al., 2020) meanwhile has extensive labels for action performed by pedestrians in real-world scenes, however unlike our dataset they do not provide any linguistic instructions for the mobility to navigate in the scene.

3 Data Collection

We discuss the details of our annotation steps for collecting data for destination prediction using action inference. We start by explaining the data preprocessing we use for data annotated with user actions. Then we explain the two annotation steps for the next action and destination prediction using action knowledge shown in Figure 2. We additionally collect information regarding the attributes present in our data detailed in the final subsection.

3.1 Data Preprocessing

We use the Titan (Malla et al., 2020) dataset as the base to reduce the total number of annotations required, as it already has action labels. The Titan dataset has videos with bounding boxes for objects and pedestrians and their actions in each frame. We chose the Titan dataset because of the high-quality frame-to-frame annotation of actions in real driving scenes. Although Titan has multiple labels, we start by using the simple contextual action, which has the most variation among the different kinds of action labels available. The Titan dataset has 12 unique actions labeled as simple contextual actions, such as *walking along the side of the road* or *exiting a building*. The process we use for making videos from the frame-to-frame data of Titan and how we filter these videos for higher quality is included in Appendix A. For the data collection, we randomly selected a maximum of 50 videos for each action type, resulting in 294 unique videos.

3.2 Next Action Annotation

In this step, we use Amazon Mechanical Turk for the annotation process. We show a short clip of a person performing a certain action to crowd workers and ask them to annotate the likely next action in the next 5 seconds after the video ends. For example, in Figure 1, a predicted next action would be *take a left and walk towards my office*. We ask the workers to always start with a verb, making action verbs and state verbs the scope of the next action annotation. We added conditions that the regions of the person should be from English-speaking countries. We also included the Amazon master certification requirement, with a minimum approval rate of 95% and a minimum of 1000 hits approved. In total, 23 unique annotators worked on the 294 videos used in this step, and each annotator was paid $0.75 per video.

3.3 Destination Prediction Annotation

In this annotation step, we show crowd workers the same clip as step 1 (next action annotation) and give them the belief of the next action in the near future collected in step 1. Given this next action, we asked the workers to mark out the correct destination, imagining that they were the taxi driver and the person highlighted in the video was the passenger they were about to pick up. For example, as in the image on the right side of Figure 2, we want the workers to come up with the destination on

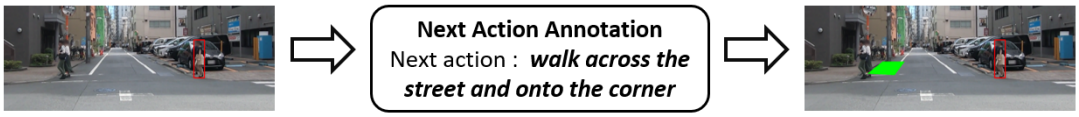

Figure 2: Proposed annotation pipeline focused on action inference.

Command Type	Transformer Model			Fully Convolutional Model		
	Accuracy	Recall@100	IOU	Accuracy	Recall@100	IOU
All action knowledge	25.3	26.7	**15.1**	22.6	23.5	12.3
Action only	**25.9**	**28.6**	11.2	24.2	15.6	11.9
Next action only	24.0	25.6	11.8	14.3	16.7	8.5
No action knowledge	18.6	19.7	10.1	21.6	22.1	11.9

Table 1: Experiment results with the two variations of models and ablation of action knowledge.

the other side of the road so that it becomes more convenient to pick up the person after they cross the road. If there is no appropriate destination, we ask the annotators to choose the option of nothing to label and give a short reason for no appropriate destination. Because of the complexity of this annotation task, we had two rounds of qualification based on the appropriateness of the destination to filter out the good annotators. We limit the final annotation to 11 good-quality annotators.

3.4 Additional Attributes

After completing the above two steps, we end up with 1944 pairs of actions and the next actions. We collect the relationship between the action and the next action for these pairs. We define that the relations between action and the next action can be of three different types namely CONSEQUENCE, INDEPENDENT, and SAME. CONSEQUENCE is when the next action can only happen as a consequence of the previous one, for example, the action *stopping in front of the building to talk to a friend* can be the consequence of the previous action *walking out of a building*. INDEPENDENT is when they occur concurrently, like *walking up by the side of the road* and *talking on the phone*. SAME is when the next action is a continuation of the action, for example when action is *crossing a street at pedestrian crossing* and next action is continue through the crosswalk to the other side of the street.

We also collected data about how the next actions affect the destination. To do this we selected two different next actions for the same user and asked the annotators to mark whether the resulting destinations have a high or low overlap. We also asked them to classify the reason for the said overlap based on the difference of the next actions into four classes. The first class *no_effects* is when the difference in the next actions has no effect over the choice of destination. Further, *similar_effect* is when both next actions have the same kind of effect on the destination, and *causes_difference* is when the difference in the next action causes the difference in the destination. The final class of *others* is when there is no clear relationship between the difference of the next actions and the overlap of the destinations. Details of the data collected have been included in Appendix B.

4 Experiments

Because of the associated task proximity, we picked up the destination prediction models executed in Talk2Car-RegSeg (Rufus et al., 2021). The authors of Talk2Car-RegSeg propose finding the destination road segment on the image that the user wants the mobility to move to based on a reference expression. They propose two model variations in their study based on the variation used on how to combine multimodal features. One variation is a Transformer-based grounding model for combining the multimodal features, and another is a fully convolutional network-based model. We use both these variations to benchmark our data by fine-tuning and testing. See detailed model settings and parameters used while fine-tuning in Appendix C.

Since the models are based on finding the destination based on reference expression, we use manually created templates to generate reference expressions based on action and next action knowledge. The advantage of templates is that we control which

Command Type	Transformer Model			Fully Convolutional Model		
	Accuracy	Recall@100	IOU	Accuracy	Recall@100	IOU
All action knowledge	**25.3**	**26.7**	**15.1**	22.6	23.5	12.3
PERSPECTIVE	24.0	26.1	13.1	21.0	21.3	10.0
ANAPHORA	20.8	22.6	13.1	24.5	27.5	12.6
PARAPHRASE	23.6	24.0	14.1	25.1	25.6	16.1

Table 2: Experiment results with grammatical variation of reference expressions.

Type	C	I	S
All action knowledge	19.4	12.4	9.1
Next action only	14.7	9.9	7.8
Action only	11.5	10.6	11.9

Table 3: IOU measures of Transformer-based model based on the relationship between action and next action. Here, C refers to CONSEQUENCE relation, I refers to INDEPENDENT, and S refers to SAME.

Overlap Level	Relation	BC	Opp	BI
High	no_effects	85	34	170
	similar_effect	29	18	93
	causes_difference	3	2	4
	others	1	0	3
Low	no_effects	3	109	116
	causes_difference	5	99	122
	similar_effect	2	8	18

Table 4: Accuracy distribution based on overlap level of destinations and reason for overlap based on different next actions. Here, BC refers to cases where prediction is correct for both cases, Opp when it is correct in one case and incorrect in another, and BI refers to both incorrect.

language phenomenon we are trying to test. The template using action and next action knowledge is *I am the person in the red box. I am* <ACTION>. *I will* <NEXT ACTION>. *Could you pick me up?*. As this may cause issues with the naturalness of the sentence, we use a grammar corrector (Damodaran, 2021) to correct the sentence. As an ablation study, we create three variations of this template, eliminating all action data or one of the action data and the next action data.

We also create templates focused on grammatical variations. PERSPECTIVE template assumes the user is inside the mobility and refers to the person using the red box. ANAPHORA template refers to the person through the pronoun. To cover a wide syntactic and semantic variety of reference expressions, we also provide sentences rephrased from the template with action and next action knowledge using GPT-4 (OpenAI et al., 2024). We call rephrased sentences as PARAPHRASE. We manually verified that the results' content corresponded to the meaning of the original reference expressions. Additional information regarding the prompt used and examples for the variations of reference expressions are given in Appendix D.

We choose a strategy of fine-tuning and testing the pre-trained versions of the models in Talk2Car-RegSeg. We split the 294 images into 236 images for seen fine-tuning data and 58 images for unseen test data. The seen split used for fine-tuning is again divided into 80% for training and 20% for validation.

We use three different comparison metrics for our experiments. Accuracy refers to the Pointing game score as defined by the authors of Talk2Car-RegSeg (Rufus et al., 2021). According to this definition, accuracy is when the point with the highest likelihood in the output mask is in the ground truth. Pointing game scores can be justified based on the general trend of autonomous mobility control algorithms being able to navigate based on single points and not needing an entire segment of navigation points. Secondly, recall@100 can be defined as whether at least one of the 100 top likelihood points is in the ground truth. Finally, for Intersection over Union (IOU) we use the standard definition of intersection area of predicted and ground truth segments divided by the union of the area of the two segments.

We modify all three of the above accuracy scores to give output one when the model correctly assigns that none of the pixels has greater than the threshold accuracy in case of no destination.

5 Results

From Table 1, for all three evaluation metrics, models fine-tuned with no action knowledge perform the worst, proving that models benefit from the presence of action knowledge in the reference expressions. An increase in accuracy with action knowledge indicates both models can learn to perform action inference. The best performance oc-

curs when using only action data, likely because it comes from the Titan dataset, with minimal linguistic variation. Comparatively for the next action, the annotators are asked to use free-form language, which makes them more complicated and confusing for models to infer. However, compared to the values stated in Talk2Car-RegSeg, the accuracy falls over 50%, showing that there is still scope for improvement.

Table 2 summarizes results for grammatical variation for expressions containing all action knowledge. We observe a drop for PERSPECTIVE and PARAPHRASE cases in the case of the Transformer-based model whereas an opposite trend in the case of the convolutional network-based model which simply averages the embeddings over the text. This indicates a lack of depth in the language branch of the Transformer-based model. We especially see a significant drop in the ANAPHORA case, which means that the models have more difficulty in generalizing the performance in the presence of anaphora.

Table 3 presents how the relationship between action and next action affects the destination accuracy for the transformer model. We observe that in the case where the next action is a consequence of the action being performed, the performance improves considerably when both action and next action knowledge are available. This suggests that such cases need reasoning with the combination of action knowledge. However, it deteriorates in the case where action and next action are classified as the same. This indicates that free-form next action knowledge has a more deteriorating effect compared to the positive effect of compounding knowledge.

Table 4 gives us insight on a case-by-case basis into how the difference between the next actions affects the results of the fine-tuned model with all action knowledge. A model could be said to be performing well on action inference if higher values are observed in both correct (BC) columns compared to the other two columns. We can see that the performance is especially low in cases with, low overlap in destinations. This leads us to believe that the model still can not learn the differences between the effects of different beliefs over the next actions.

6 Conclusion

In this work, we created a new VLN destination prediction dataset for a vehicle to pick up a pedestrian. This dataset focuses on how the actions of the pedestrian and the next actions the pedestrian is likely to do in the near future affect the destination of the vehicle, which we define as action inference. We also provide attributes of the relationship between the action and the next action along with reasoning about how the difference in the next actions affects the overlap of destinations. Our experiments on fine-tuning pre-trained destination prediction models resulted in a higher action inference accuracy when action knowledge is present in the instruction phrase. This indicates that models can learn to reason about action knowledge to a certain extent. However, we see a drop of 50% when we compare the test accuracy on our dataset compared to the test accuracy on the pre-training dataset. We also observe underwhelming performance in cases where the next action variation causes a low overlap of destinations. Both of these results lead us to believe that there is still scope for improvement. In our future work, we would like to work on creating architectures that could better handle action inference.

Limitations

This work focuses on collecting data regarding action inference for destination prediction. However, in real driving scenes, our dataset is still limited as it does not collect data on traffic rules affecting the destination during each situation. To accurately come up with all the factors taken into the reasoning for determining the destination is a difficult task. However, there is still scope for more factors that could have been easily added to this data collection such as events occurring in the scene or social situation of the user. Also,since destination prediction is a field that has not been explored in much depth, there are few models against which we can benchmark our dataset. We select Talk2Car-RegSeg as our baseline as also work on destination prediction.

Acknowledgments

We thank the three anonymous reviewers for their helpful comments and feedback. This work was partially supported by JST, PRESTO Grant Number JPMJPR21C8, Japan.

References

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. TOUCHDOWN: natural language navigation and spatial reasoning in visual street environments. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12538–12547. Computer Vision Foundation / IEEE.

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 40(04):834–848.

Prithiviraj Damodaran. 2021. Gramformer. https://github.com/PrithivirajDamodaran/Gramformer.

Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. 2019. Talk2Car: Taking control of your self-driving car. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2088–2098, Hong Kong, China. Association for Computational Linguistics.

Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. 2020. Learning to follow directions in street view. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11773–11781.

Srikanth Malla, Behzad Dariush, and Chiho Choi. 2020. TITAN: future forecast using action priors. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11183–11193. IEEE.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Bar-

ret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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, Doha, Qatar. Association for Computational Linguistics.

Nivedita Rufus, Kanishk Jain, Unni Krishnan R Nair, Vineet Gandhi, and K Madhava Krishna. 2021. Grounding linguistic commands to navigable regions. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 8593–8600. IEEE Press.

Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. 2021. Talk2nav: Long-range vision-and-language navigation with dual attention and spatial memory. *Int. J. Comput. Vision*, 129(1):246–266.

A Further Details of Data Preprocessing

In this section, we describe how we filtered out the data generated from Titan (Malla et al., 2020). We first highlight the person performing an action by using a red bounding box around the person. We then truncate the video based on the frames in which a person is present. We clear out cases where a person performs more than one action in the video, performing different actions across frames. We also filtered out the videos that were too short, where the person is visible for less than 30 frames, and where the person's bounding box size is too small.

B Analysis of Dataset Contents

Table 5 represents the attributes of the action knowledge we have collected. Due to the abundance of CONSEQUENCE and INDEPENDENT relations our dataset can only solved by the reasoning based on a combination of action and next action. Table 5 represents how two different next actions affect the destination prediction. We can see that a high overlap of destinations is caused when the next actions have a similar effect. In addition, a low overlap of destinations is caused by the difference in the effects of the next actions. The greater than expected where differences in the next action have no effect can be explained by the presence of a high number of INDEPENDENT next action cases.

C Details of Models and Fine-tuning

According to the setup followed in Talk2Car-RegSeg (Rufus et al., 2021), for both the transformer and fully convolutional model we use

Action	C	I	S
waiting to cross street	30	35	10
walking along the side of the road	143	122	62
walking on the road	126	147	61
crossing a street at pedestrian crossing	184	106	35
jaywalking	194	101	34
biking	108	161	61
motorcycling	34	40	9
cleaning an object	36	17	17
closing	3	2	2
entering a building	20	9	7
exiting a building	17	3	0
opening	6	1	0
Total	901	744	298

Table 5: Number of next action for each action and relationship type. Here, C refers to CONSEQUENCE relation, I refers to INDEPENDENT, and S refers to SAME.

Overlap Level	Relation	Count
High	no_effects	1593
	similar_effect	779
	causes_difference	43
	others	41
Low	no_effects	1211
	causes_difference	1257
	similar_effect	169
	others	6

Table 6: Summary of overlap level of destination and reason of overlap for different next actions for the same user.

DeeplabV3 (Chen et al., 2018) with ResNet_101 backbone as image encoder to extract the visual features. 300d GloVe embeddings pre-trained on Common Crawl 840B tokens (Pennington et al., 2014) are used to embed the reference expressions. In the case of the fully convolutional model, the embeddings of the reference expressions are averaged and then appended to the image embedding along the channel dimension to form the multimodal feature used for learning. This setting according to the user loses the sequential word information because of the averaging step. For the Transformer-based model, the word embeddings of the reference expression are appended along the length dimension of the image embedding and the self-attention feature of the transformer enables multimodal learning across the embeddings.

For fine-tuning we use a batch size of 64. Since compared to the reference expression used in our

case being longer than the ones present in the original Talk2Car-RegSeg dataset, we increase the maximum token length of the expression to 200 while finetuning. As a loss function, we use Dice loss as it penalizes more harshly on IoU thereby leading to better learning in our dataset. We use the AdamW optimizer with a learning rate of 0.0003 with a weight decay of 0.005. We observed that across all the templates both the transformer and fully convolutional models tend to converge around 20 epochs.

D Variations of Reference Expressions

In this section, we explain the different variations of reference expressions generated by us using the action and next action as shown in Figure 2. In this case, the action is *walking along the side of the road* and the next action would be *walk across the street and onto the corner*. The following will be the phrases generated for the ablation study and template-based grammatical variations with all action knowledge.

- All action knowledge: I am the person in red box. I am walking along the side of the road. I will walk across the street and onto the corner. Could you pick me up?
- Action only: I am the person in red box. I am walking along the side of the road. Could you pick me up?
- Next action only: I am the person in red box. I will walk across the street and onto the corner. Could you pick me up?
- No action knowledge: I am the person in red box. Could you pick me up?
- PERSPECTIVE: There is a person in red box. The person is walking along the side of the road. The person is about to walk across the street and onto the corner. Could you pick the person up?
- ANAPHORA: There is a person in red box. He is walking along the side of the road. He is about to walk across the street and onto the corner. Could you pick him up?

For PARAPHRASE we use the following to prompt GPT-4 (OpenAI et al., 2024):

- Input system message: You are a phrase generator asked to rephrase an expression used for a vision and language task. You will rephrase in such a way that the original meaning and storyline flow in the phrase is still unchanged. Answer should only be the rephrased sentence, please do not use any extra words.
- Input prompt message: I am the person in red box. I am walking along the side of the road. I will walk across the street and onto the corner. Could you pick me up?
- Output paraphrased response: Could you come get me? I'm the individual encased in a red square, taking a stroll by the roadside. I plan to cross the road and reach the corner.