RUN through the Streets:
A New Dataset and Baseline Models for Realistic Urban Navigation

Tzuf Paz-Argaman[1] and **Reut Tsarfaty**[1,2]
[1]Open University of Israel
[2]Allen Institute for Artificial Intelligence
{tzufar,reut.tsarfaty}@gmail.com

Abstract

Following navigation instructions in natural language requires a composition of language, action, and knowledge of the environment. Knowledge of the environment may be provided via visual sensors or as a symbolic world representation referred to as a *map*. Here we introduce the *Realistic Urban Navigation* (RUN) task, aimed at interpreting navigation instructions based on a real, dense, urban map. Using Amazon Mechanical Turk, we collected a dataset of 2515 instructions aligned with actual routes over three regions of Manhattan. We propose a strong baseline for the task and empirically investigate which aspects of the neural architecture are important for the RUN success. Our results empirically show that *entity abstraction*, *attention over words and worlds*, and a constantly updating *world-state*, significantly contribute to task accuracy.

1 Introduction and Background

The task of interpreting and following natural language (NL) navigation instructions involves interleaving different signals, at the very least the linguistic utterance and the representation of the world. For example, in "turn right on the first intersection", the instruction needs to be interpreted, and a specific object in the world (the intersection) needs to be located in order to execute the instruction. In NL navigation studies, the representation of the world may be provided via visual sensors (Misra et al., 2018; Blukis et al., 2018; Nguyen et al., 2018; Yan et al., 2018; Anderson et al., 2018) or as a symbolic world representation. This work focuses on navigation based on a symbolic world representation (referred to as a *map*).

Previous datasets for NL navigation based on a symbolic world representation, HCRC (Anderson et al., 1991; Vogel and Jurafsky, 2010; Levit and Roy, 2007) and SAIL (MacMahon et al., 2006; Chen and Mooney, 2011; Kim and Mooney, 2012, 2013; Artzi and Zettlemoyer, 2013; Artzi et al., 2014; Fried et al., 2017; Andreas and Klein, 2015) present relatively simple worlds, with a small fixed set of entities known to the navigator in advance. Such representations bypass the great complexity of real urban navigation, which consists of long paths and an abundance of previously unseen entities of different types.

In this work we introduce *Realistic Urban Navigation* (RUN), where we aim to interpret navigation instructions relative to a rich symbolic representation of the world, given by a real dense urban map. To address RUN, we designed and collected a new dataset based on OpenStreetMap, in which we align NL instructions to their corresponding routes. Using Amazon Mechanical Turk, we collected 2515 instructions over 3 regions of Manhattan, all specified (and verified) by (respective) sets of humans workers. This task raises several challenges. First of all, we assume a large world, providing long routes, vulnerable to error propagation; secondly, we assume a rich environment, with entities of various different types, most of which are unseen during training and are not known in advance; finally, we evaluate on the full route intended, rather than on last-position only.

We then propose a strong neural baseline for RUN where we augment a standard encoder-decoder architecture with an entity abstraction layer, attention over words and worlds, and a constantly updating world-state. Our experimental results and ablation study show that this architecture is indeed better-equipped to treat *grounding* in realistic urban settings than standard sequence-to-sequence architectures. Given this RUN benchmark, empirical results, and evaluation procedure, we hope to encourage further investigation into the topic of interpreting NL instructions in realistic and previously unseen urban domains.

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing*, pages 6449–6455,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

	Symbolic World Representation				NL Utterances							
Map	**Tokens in Map**	**Entities in Map**	**Number of Tiles**	**Size (km^2)**	**Paragraphs**	**Instructions**	**Unique Tokens in Paragraphs**	**Avg. Tokens per Instruction**	**Avg. Verbs per Instruction**	**Avg. Actions per Instruction**	**Instruction with advmode before ROOT**	**Avg. Named Entities per Instruction**
1	10,353	1,612	5,457	0.51	159	874	735	14.42	1.53	18.71	12.81%	1.93
2	9,829	1,134	4,935	0.46	128	884	727	12.46	1.31	12.81	9.16%	1.32
3	8,844	1,051	5,452	0.51	102	757	654	11.86	1.29	12.44	10.3%	1.44
Corpus	29,026	3,797	15,844	1.48	389	2515	**1451**	12.96	1.38	13.71	10.78%	1.57

Table 1: Data Statistics of RUN: statistics over different maps and the full corpus. The table is divided into features of the symbolic world representation and the written paragraphs.

Phenomenon	instructions	Example from RUN
Reference to unnamed entity	53.33%	Walk to the first **stoplight** and turn left heading south.
Reference to named entity	93.33%	Walk a little more and you will reach your destination on your right: **Fantastic Cafe**.
Coreference	10%	Pass the intersection and **it** will be the second building on your right.
Sequencing	20%	Walk to the **next** intersection, turning right to Avenue B.
Count	23.33%	Walk **5** buildings down the street, and you will see the mcdonalds.
Egocentric spatial relation	26.67%	B&H photo will be immediately **on your right** and that is where you want to be.
Imperative	83.33%	**Go** through the intersection and **follow** the road past the Kmart center.
Direction	66.67%	Make a **right** and go up one block to the light on West 33rd street and 7th Avenue .
Condition	26.67%	we will continue walking **till** we come to the east 7th street intersection.
Verification	20%	On your left toward about the middle of the block **you'll see** Alphabet City.

Table 2: Linguistic Analysis of RUN: we analyze 30 randomly sampled instructions in RUN. The table characterizes linguistic phenomena in RUN, categorized according to the catalogue presented in Chen et al. (2018).

	#Entities	#Unique Entities	#Tiles	Tiles Moved per Sentence\Paragraph
HCRC	*11.93	*8.125	*11.93 [1]	n/a*9.75
SAIL	22	0	33.33	1.3\5.34
TTW	*62.6	**0	25	n/a\2.5
RUN	932	365	1059.6	12.2\78.89

Table 3: Quantitative Comparison of the HCRC (Anderson et al., 1991), SAIL (MacMahon et al., 2006), TTW (de Vries et al., 2018), and the new RUN Dataset. *We average over three randomly chosen maps. **de Vries et al. (2018) assume perfect perception: all entities at each location are known in advance.

2 The RUN Task and Dataset

In this work we address the task of following a sequence of NL navigation instructions given in colloquial language based on a dense urban map.

The *input* to the RUN task we define is as follows: (i) a map with rich details divided into tiles, (ii) an explicit starting point, and (iii) a sequence of navigation instructions we henceforth refer to as a navigation *paragraph* . We refer to each sentence as an *instruction*, and we assume that following the individual instructions in the *paragraph* one by one will lead the agent to the intended end-point. The *output* of RUN is the entire route described by the paragraph, i.e., all coordinates up to and including its end-point, pinned on the map.

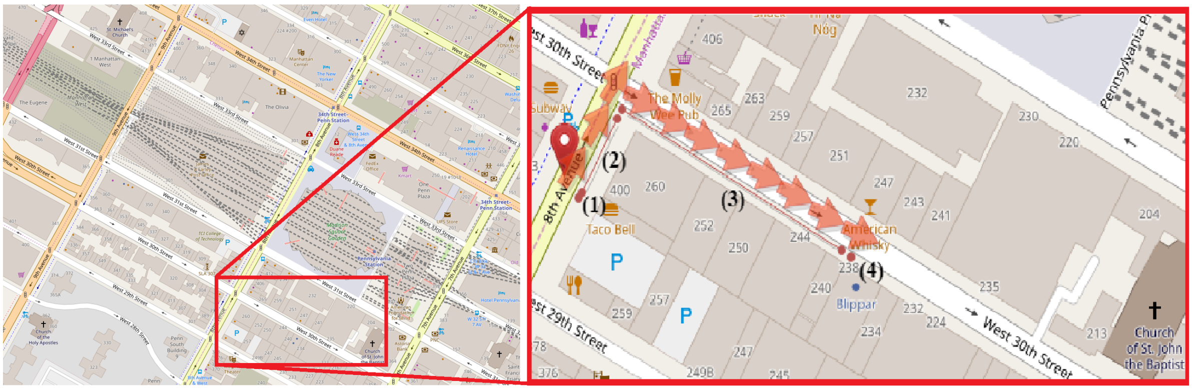
In order to address RUN we designed and collected a novel dataset, henceforth the *RUN dataset*, which is based on OpenStreetMap (OSM).[2] The map contains rich layers and an abundance of entities of different types. Each entity is complex and can contain (at least) four labels: name, type, is_building=y/n, and house number. An entity can spread over several tiles. As the maps do not overlap, only very few entities are shared among them. The RUN dataset aligns NL navigation instructions to coordinates of their corresponding route on the OSM map.

We collected the RUN dataset using Amazon Mechanical Turk (MTurk), allowing only native English speakers to perform the task. We collected instructions based on three different areas, all in urban, dense parts of Manhattan. The size of each map is 0.5 km^2. The dataset contains 2515 navigation instructions (equal to 389 complete paragraphs) paired with their routes. The paragraphs are crowd-sourced from 389 different instructors, of which style and language vary (Geva et al., 2019).

Our data collection protocol is as follows. First, we asked the MTurk worker to describe a route between two landmarks of their choice. After having described the complete route in NL, the same worker was instructed to pin their described route on the map. This was moderated by showing them the paragraph they narrated, sentence by sentence,

[1] The task defined by Vogel and Jurafsky (2010) is of moving between entities only.

[2] OSM is a free, editable, map of the whole world, that was built by volunteers, with millions of users constantly adding informative tags to the map.



Instructions: (1) As you walk out of Taco Bell on 8th Avenue, turn right. (2) Then turn right as you reach the intersection of West 30th Street. (3) Now head down West 30th Street for approximately a half block. (4) You have gone too far if you reach Church of St. John the Baptist.

Figure 1: An Example of a Short Navigation Paragraph: showing a navigation paragraph at the bottom and two images - full map (left image) and a small part of the map (right image). In the full map, many entities are not seen until zoom-in is applied. The navigation paragraph is divided into four sentences: (1) sentence requires a turn action; (2) requires implicit walk actions and an explicit turn; (3) requires walk actions; the last (4) sentence is a verification only and no action is required.

so that they have to pin on the map each instruction separately. A worker could only pin routes on street paths. Furthermore, on every turn the worker had to mark an explicit point on the map which marked the direction in which the worker needs to move next. An example of simple individual instructions and their respective route is given in Figure 1.

We then asked a disjoint set of workers (testers) to verify the routes by displaying the starting point of the route, and displaying the instructions in the paragraph sentence-by-sentence. The tester had to pin the final point of the sentence. Each route was tested by three different workers. Testing the routes allowed us to find incorrect routes (paragraphs that don't match an actual path) and discard them. They also provide an estimate of the human performance on the task (Reported in Section 4, Experiments).

Having collected the data, we divided the map into tiles, each tile is 11.132 m X 11.132 m. Each tile contains the labels of the entities it displays on the map, such as restaurants, traffic-lights, etc., and the walkable streets in it. Each walkable street is composed of an ordered list of tiles, including a starting tile and an end tile. Table 1 shows statistics over the dataset. Table 2 characterize linguistic phenomena in RUN, categorized according to the catalogue of Chen et al. (2018). Table 3 shows a quantitative comparison of the RUN dataset to previous datasets of map-based navigation. The table underscores some key features of RUN, relative to the previous tasks. RUN contains longer paths and many more entities that are unique, thus appearing for the first time during testing; the size of the map is on a different scale than previous tasks, thus, amplifying the grounding challenge; the number of tiles moved is accordingly larger than in previous datasets, hence increasing the vulnerability to error propagation.

Overall, RUN contains challenging linguistic phenomena, at least as in previous work, and a rich environment, with more realistic paths than in previous tasks.

3 Models for RUN

We model RUN as a sequence-to-sequence learning problem, where we map a sequence of instructions to a sequence of actions that should be performed in order to pin the actual path on the map. The execution system we provide for the task defines three types of actions: 'TURN', 'WALK', 'END'.[3] 'TURN' is one of the following: right-turn, left-turn, turn-around. The turning move is not necessarily an exact 90-degree turn; the execution system looks for the closest turn option.

[3] RUN has a variation that contains two more types of actions: 'FACE' is a change of direction to face a specific street and end of the street; 'VERIFY' gives verification to the current direction when it is explicitly mentioned in the instructions. For example, "turn right on to 8th Avenue". However, we have been unsuccessful in benefiting from "FACE" and "VERIFY". We leave them for future research.

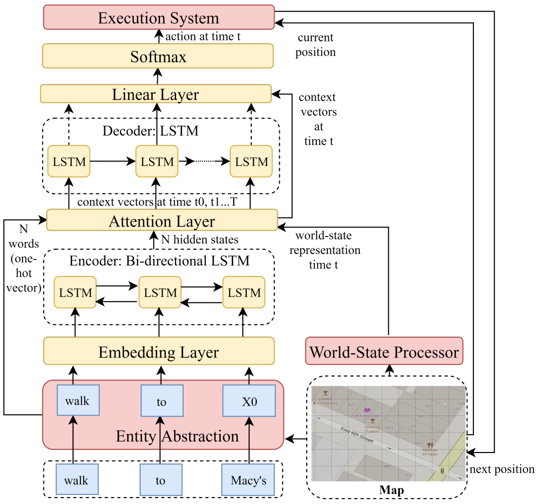


Figure 2: Our Model, Conditioned Generation with Attention over Words and World-States, an Entity Abstraction Layer and an Execution System (CGAEW). The light color (yellow) parts presents a standard Encoder-Decoder with attention, while the dark color (red) are components added on top of a standard CGA.

'WALK' is a change of position in the direction we are facing. The streets can be curved, so 'WALK' is relative to the street that the agent is on. Each street is an ordered-list of tiles, so an action of walking two steps is in fact two actions of 'WALK', in the direction the agent is facing. The 'END' action defines the end of each route. The input consists of an instruction sequence $x_{1:N}$, a map M, and a starting point p_0 on the map. The output is a sequence of actions $a^*_{1:T}$ to be executed.

$$a^*_{1:T} = arg \max_{a_{1:T}} P(a_{1:T}|x_{1:N}, M, p_0)$$

$$= \arg\max_{a_{1:T}} \prod_{t=1}^{T} P(a_t|a_{1:t-1}, x_{1:N}, M, p_0)$$

Where x_i denote sentences, a_i denotes actions, M is the map and p_0 is the starting point.

Our basic model for RUN is a sequence-to-sequence model similar to the work of Mei et al. (2015) on SAIL, and inspired by Xu et al. (2015). It is based on Conditioned Generation with Attention (CGA). To this model we added an Entity abstraction layer (CGAE) and a World-state representation (CGAEW). It thus consists of six components we describe in turn – Encoder, Decoder, Attention, Entity Abstraction, World-State Processor, Execution-System. The complete architecture is depicted in Fig. 2.

The Encoder takes the sequence of words that assembles a single sentence and encodes it as a vector using a biLSTM (Graves and Schmidhuber, 2005). **The Decoder** is an LSTM generating a sequence of actions that the execution-system can perform, according to weights defined by an Attention layer. The **Entity Abstraction** component deals with out-of-vocabulary words (OOV). We adopt a similar approach to Iyer et al. (2017); Suhr et al. (2018), replacing phrases in the sentences which refer to previously unseen entities with variables, prior to delivering the sentence to the Encoder. E.g., "Walk from Macy's to 7th street" turns into "Walk from X1 to Y1". Variables are typed (streets, restaurants, etc.) and are numbered based on their order of occurrence in the sentence. The numbering resets after every utterance, so the model remains with a handful of typed entity-variables. The **World-State Processor** maps variables to the entities on the map which are mentioned in the sentence. The world-state representation consists of two vectors, one representing the entities at the current position, and one representing the entities in the path ahead. The **Attention** layer considers the sequence of encoded words *as well as* current world-state, and provides weights on the words for each of the decoder steps. In both training and testing, the **Execution-System** executes each action separately to produce the next position.[4]

4 Experiments

We evaluate our model on RUN and assess the contribution of the particular components that we added on top of the standard CGA model.

We train the model using a negative log-likelihood loss, and used Adam (Kingma and Ba, 2014) optimization. For weights initialization we rely on Glorot and Bengio (2010). We used a grid search to validate the hyper-parameters. The model converged at around 30 epochs and produced good results with 0.9 drop-out and a beam of size 4. During inference, we seek the best candidate path using beam-search and normalize the scores of the sequences according to Wu et al. (2016).

We follow the evaluation methodology defined by Chen and Mooney (2011) for SAIL where we use three-fold validation, and in each fold, we use two maps for training (90%) and validation (10%) and test on the third one. We report a sized-

[4] Our code, models, complete maps, annotated dataset, and evaluation: https://github.com/OnlpLab/RUN.

NO-MOVE	RANDOM	JUMP	CGA	CGAE	CGAEW	HUMAN
30.3\0.3	11.2\0.1	26.3\0	43.68 (5.93)\0.26	46.01 (6.13)\6.17	62.37 (3.11)\10.45	n/a\81.12

Table 4: Bounds on Accuracy for Sentences\Paragraphs, weighted averages over folds (std).

	Sentence	JUMP baseline	CGA	CGAE	CGAEW
1	Just before 9th Avenue, you will see your destination on the right, the West Side Jewish Center.	✓	✗	✗	✓
2	Turn left onto West 34th Street.	✗	✓	✓	✓
3	At the 8th Avenue and West 20th Street intersection, turn right onto West 20th Street.	✗	✗	✓	✓
4	Keep going till you get to the intersection of West 21st Street.	✗	✗	✗	✓
5	Head west on East 7th for 2 (large) blocks; Its a one-way street.	✗	✗	✗	✗

Table 5: Error analysis of all models, for different instructions, showing succeeded / failure on predicting the path.

weighted average test result. For all models we report the accuracy per single sentences and full paragraphs. Success is measured by generating an exact route, not striding away from the path. The last position on the path should be within five tile euclidean distance from the intended destination, as the position explained in the instruction might not be specific enough for one tile.[5] In single-sentences, the last position should also be facing the correct direction.

We provide three simple baselines for the RUN task: (1) NO-MOVE: the only position considered is the starting point; (2) RANDOM: As in Anderson et al. (2018), turn to a randomly selected heading, then execute a number of 'WALK' actions of an average route; (3) JUMP: at each sentence, extract entities from the map and move between them in the order they appear. If the 'WALK' action is invalid we take a random 'TURN' action.

Table 4 shows the results for the baseline models as well as the HUMAN measured performance on the task. The human performance provides an upper bound for the RUN task performance, while the simple baselines provide lower bounds. The best baseline model is NO-MOVE, reaching an accuracy of 30.3% on single sentences and 0.3 on complete paragraphs. For the HUMAN case, paragraph accuracy reaches above 80.

Table 4 shows the results of our model as an ablation study, and Table 5 shows typical errors of each variant. We see that CGAE outperforms CGA, as the swap of entities with variables lowers the complexity of the language that the model needs to learn, allowing the model to effectively cope with unseen entities at test time. We further found that, in many cases, CGAE produces the right type of action, but it does not produce enough of it to reach the intended destination. We attribute these errors to the absence of a world-state representation, resulting in an incapability to ground instructions to specific locations. CGAEW improves upon CGAE as the existence of world-state in the score of the attention layer allows the model better learn the grounding of entities in the instruction to the map. However our best model still fails on features not captured by our world-state: abstract unmarked entities such as blocks, intersections, etc, and generic entities such as traffic-lights (Tab. 5).

5 Conclusion

We introduce RUN, a new task and dataset for NL navigation in realistic urban environments. We collected (and verified) NL navigation instructions aligned with actual paths, and propose a strong neural baseline for the task. Our ablation studies show the significant contribution of each of the components we propose. In the future we plan to extend the world-state representation, and enable the model to ground *generic* and *abstract* concepts as well. We further intend to add additional signals, for instance coming from vision (cf. Chen et al. (2018)), for more accurate localization.

6 Acknowledgments

We thank Yoav Goldberg and Yoav Artzi, for their advice and comments. We thank the ONLP team at the Open University of Israel, for fruitful discussions throughout the process. We further thank the anonymous reviewers for their helpful comments. This research was presented at Georgetown University, Cornell University, and Tel-Aviv University. This research is supported by European Research Council, ERC-StG-2015 scheme, Grant number 677352, and by the Israel Science Foundation (ISF), Grant number 1739/26, for which we are grateful.

[5] We selected this distance as it was the average distance our successful mechanical tester-workers arrived from the intended pinned point.

References

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The hcrc map task corpus. *Language and speech*, 34(4):351–366.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018. 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.

Jacob Andreas and Dan Klein. 2015. Alignment-based compositional semantics for instruction following. *arXiv preprint arXiv:1508.06491*.

Yoav Artzi, Dipanjan Das, Slav Petrov, et al. 2014. Learning compact lexicons for ccg semantic parsing. In *EMNLP*, pages 1273–1283.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.

Valts Blukis, Nataly Brukhim, Andrew Bennett, Ross A Knepper, and Yoav Artzi. 2018. Following high-level navigation instructions on a simulated quadcopter with imitation learning. *arXiv preprint arXiv:1806.00047*.

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *AAAI*, volume 2, pages 1–2.

Howard Chen, Alane Shur, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2018. Touchdown: Natural language navigation and spatial reasoning in visual street environments. *arXiv preprint arXiv:1811.12354*.

Daniel Fried, Jacob Andreas, and Dan Klein. 2017. Unified pragmatic models for generating and following instructions. *arXiv preprint arXiv:1711.04987*.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. *arXiv preprint arXiv:1908.07898*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

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

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. *arXiv preprint arXiv:1704.08760*.

Joohyun Kim and Raymond J Mooney. 2012. Unsupervised pcfg induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 433–444. Association for Computational Linguistics.

Joohyun Kim and Raymond J Mooney. 2013. Adapting discriminative reranking to grounded language learning. In *ACL (1)*, pages 218–227.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Michael Levit and Deb Roy. 2007. Interpretation of spatial language in a map navigation task. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(3):667–679.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: Connecting language, knowledge, and action in route instructions. *Def*, 2(6):4.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2015. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. *arXiv preprint arXiv:1506.04089*.

Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping instructions to actions in 3d environments with visual goal prediction. *arXiv preprint arXiv:1809.00786*.

Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. 2018. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. *arXiv preprint arXiv:1812.04155*.

Alane Suhr, Srinivasan Iyer, and Yoav Artzi. 2018. Learning to map context-dependent sentences to executable formal queries. *arXiv preprint arXiv:1804.06868*.

Adam Vogel and Dan Jurafsky. 2010. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 806–814. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1807.03367*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

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.

Claudia Yan, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi. 2018. Chalet: Cornell house agent learning environment. *arXiv preprint arXiv:1801.07357*.