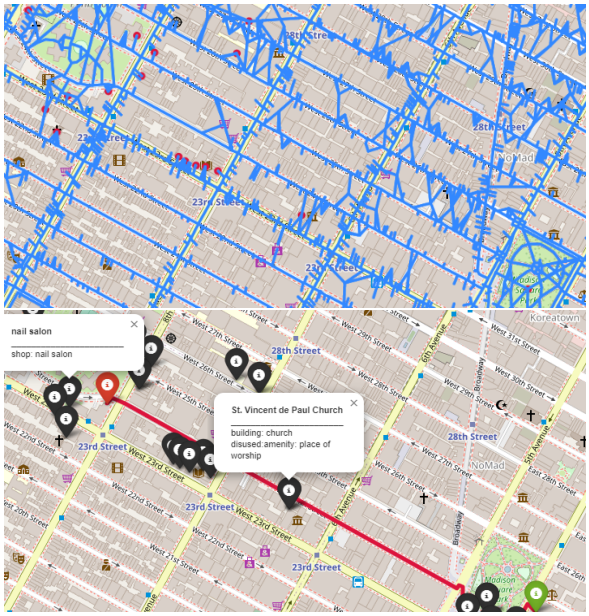
Into the Unknown: Generating Geospatial Descriptions for New Environments

Tzuf Paz-Argaman[1], John Palowitch[2], Sayali Kulkarni[2],
Reut Tsarfaty[1,2], and Jason Baldridge[2]

[1]Bar-Ilan University, [2]Google Research,
{tzuf.paz-argaman, reut.tsarfaty}@biu.ac.il,
{palowitch, sayali, jasonbaldridge}@google.com

Abstract

Similar to vision-and-language navigation (VLN) tasks that focus on bridging the gap between vision and language for embodied navigation, the new *Rendezvous* (RVS) task requires reasoning over allocentric spatial relationships (independent of the observer's viewpoint) using non-sequential navigation instructions and maps. However, performance substantially drops in new environments with no training data. Using opensource descriptions paired with coordinates (e.g., Wikipedia) provides training data but suffers from limited spatially-oriented text resulting in low geolocation resolution. We propose a large-scale augmentation method for generating high-quality synthetic data for new environments using readily available geospatial data. Our method constructs a *grounded knowledge-graph*, capturing entity relationships. Sampled entities and relations ("shop north of school") generate navigation instructions via (i) generating numerous templates using context-free grammar (CFG) to embed specific entities and relations; (ii) feeding the entities and relation into a large language model (LLM) for instruction generation. A comprehensive evaluation on RVS, showed that our approach improves the 100-meter accuracy by 45.83% on unseen environments. Furthermore, we demonstrate that models trained with CFG-based augmentation achieve superior performance compared with those trained with LLM-based augmentation, both in unseen and seen environments. These findings suggest that the potential advantages of explicitly structuring spatial information for text-based geospatial reasoning in previously unknown, can unlock data-scarce scenarios.



Meet me in the garden. Head northwest from St. Vincent de Paul Church for 2 intersections. The garden is right next to a fast-food restaurant. If you reach a nail salon, you have gone too far.

Figure 1: Our method for generating spatial descriptions samples from the graph-map (top) a path (middle image, red line), a starting point (green marker), a goal point (red marker), and prominent landmarks (black markers). It then generates an instruction (bottom) from the spatial relations between these entities.

1 Introduction

The ability to extract locations and paths from natural language descriptions of spatial information holds immense significance. This capability proves crucial in daily and disaster response scenarios, aiding the billions globally lacking formal addresses (UPU, 2012; Abebrese, 2019; Hu et al., 2023), and enhancing Geographic Information Retrieval (GIR), particularly leveraging web-based resources (Spink et al., 2002; Sanderson and Kohler, 2004).

Echoing the vision-and-language navigation (VLN) task's goal, of bridging the gap between visual perception and natural language (NL) instructions for embodied agents (Ku et al., 2020), the recently introduced Rendezvous (RVS) navigation task (Paz-Argaman et al., 2024) seeks to achieve a similar connection, but specifically be-

Findings of the Association for Computational Linguistics ACL 2024, pages 2259–2273

tween map representations and natural language NL. While VLN focuses on navigating within an environment based on visual cues and sequential instructions, RVS emphasizes the ability to utilize map information and non-sequential, often allocentric language descriptions to reach a specific target location. This shift from vision-centric instructions to map-aided guidance presents unique challenges, including reasoning about multiple spatial relationships simultaneously, inferring implicit actions from language, and navigating without explicit verification or step-by-step instructions.

However, there is a substantial gap between current models and the human performance on the RVS task, particularly in new environments that lack human annotated data. One approach to address this issue is to leverage naturally-occurring open-source data such as Wikipedia. However, these sources lack direct spatial information, which can result in models' low performance (Solaz and Shalumov, 2023). Synthesizing data using large language models (LLMs) is a common method for addressing data scarcity in NLP (Yoo et al., 2021; Edwards et al., 2021). However, for multimodal scenarios requiring precise spatial relationships, accurately generating such data without introducing errors or "hallucinations" (i.e., spurious relationships or entities), which severely undermines the performance on the underlying downstream task, remains a significant challenge.

We propose a method for generating high-quality synthetic data for new environments using open-source geospatial data (Figure 1). Our method constructs a grounded knowledge-graph of the environment, capturing spatial relationships between entities. By sampling these entities, abstract shapes like 'blocks' (implicitly derived from street relationships), and relations (e.g., 'the garden is next to a restaurant'), we generate navigation instructions by either (i) creating a large amount of templates via a generative context-free grammar (CFG), in which we embed the precise entities and relations sampled, or (ii) feeding the entities and relation into an LLM which generates the instructions.

Extensive evaluation on the RVS dataset demonstrates the clear advantage of our CFG-based method compared to the LLM approach. When navigating unseen environments, our method achieves a remarkable 9.1% absolute increase in 100-meter accuracy and a substantial 39-meter decrease in median distance error. Overall, our method helps close the human-AI performance gap in unseen environments by 45.83% in 100m accuracy and a decrease of 1,183m in median distance error. For the seen environment, our method results in an absolute improvement of 19.56% in 100m accuracy and a decrease of 151m in median distance error.

2 The Task

The Rendezvous (RVS, Paz-Argaman et al., 2024) task evaluates a system's ability to follow human-generated, colloquial language navigation instructions within a dense urban environment depicted by a map. The system is provided with three inputs: (i) Detailed Map as Knowledge Graph: a comprehensive map of the environment represented as a knowledge graph. This graph encodes spatial relationships between landmarks and other relevant features. (ii) Explicit Starting Point (Geo-coordinates): the starting location specified as a latitude and longitude coordinate pair. (iii) Navigation Instruction: a natural language instruction describing the target location's relative position to landmarks and the starting point. This instruction leverages colloquial language typically used in navigation scenarios. The RVS task demands the system to process these inputs and generate the goal location's coordinates within the defined map boundaries.

3 Proposed: Relational Augmentation

Our augmentation method aims to generate natural language location descriptions that are both accurate and well grounded. To that end, we leverage OpenStreetMap (OSM).[1] It involves three stages: (i) sampling paths; (ii) calculating spatial relations between entities; and (iii) generating instructions based on the spatial relations calculated in stage (ii). We use the OSM-based graph provided in RVS for the first and second parts and provide two methods for generating instructions based on the spatial relations between entities calculated in that part.

3.1 Sampling paths

To generate RVS-like samples accurately, we follow the RVS sampling protocol. We randomly sample small entities for the end point (the entity's shape has a maximum radius of 100 meters). For the start point, we randomly select an entity that (i) is within 200–2000 meters of the end point; (ii) has a name or type tag (e.g., a bookshop). This

[1] OpenStreetMap is a user-updated map of the world – http://www.openstreetmap.org

garden | St. Vincent de Paul Church | 2

Meet me in the ***GOAL_TYPE***. Head northwest from ***LANDMARK_ALONG_PATH*** for ***NUMBER_INTERSECTIONS*** intersections.

garden | fast-food restaurant | nail salon

The ***GOAL_TYPE*** is right next to a ***NEAR_LANDMARK***. If you reach a ***BEYOND_PIVOT***, you have gone too far.

Figure 2: Instruction generation steps (for example presented in Figure 1): (i) Template creation via CFG; (ii) replacing generic elements (in capital letters) with specific landmarks and spatial relations (above the lines).

information allows us to refer to the entity in the instruction by its definite description and not by its proper name. Finally, we pick a route that is the shortest distance between the start and end points.

3.2 Calculating Spatial Relations between Entities

Leveraging the Open Street Map (OSM) graph, we identify prominent landmarks relevant to the sampled paths for inclusion in navigation instructions. However, not all paths necessarily contain all types of landmarks and features. For instance, if the navigation ends in a dead-end street, landmarks beyond the leading path might not be relevant to the task. This section details the criteria for landmark selection and the spatial relations computed with respect to the start and end points, and the path.

Picking landmarks We pick three different types of landmarks with a certain spatial relation that will be referenced in the instruction: (i) landmarks close to the end point (within 100 meters from it); (ii) landmarks along the route; and (iii) landmarks that are on the same street as the goal location but beyond the route, such that if the agent keeps on walking beyond the goal, it will reach that landmark ("beyond landmark"). Priority for landmark selection is given to those with the highest level of external recognition, as determined by the following hierarchy: has a Wikipedia or Wikidata[2] link, is a brand, is a tourism attraction, is an amenity, is a shop. We randomly select all landmarks from the most prominent level found. If in the area there are multiple landmarks of the same type, we group the landmarks according to their type and quantity (e.g., 'two book shops'). If a landmark is far (over 200 meters) from the end point and it has a proper name, we can use its proper name in the text generation, e.g., 'the Empire State Building'. If it is near (less than 200 meters) the end point, we will always use the indefinite name of the landmark, e.g., 'a bookshop'.

Calculating spatial relations The objective is to determine the spatial relationship between landmarks and the end point. There are several ways to describe the relationship between two entities, such as the number of blocks between them. The spatial relations calculated are (i) allocentric relations, i.e., cardinal directions, between landmarks, start and end points. Cardinal direction is calculated by the bearing θ_{degrees} between two points (longitude, latitude), (x_1, y_1) and (x_2, y_2):

$$\begin{aligned}\theta_{\text{radians}} = (&\tan^{-1}(\sin(\lambda_2-\lambda_1)\cos(\varphi_2),\\ &\cos(\varphi_1)\sin(\varphi_2))\\ &-\sin(\varphi_1)\cos(\varphi_2)\cos(\lambda_2-\lambda_1))\\ &+360^\circ) \mod 360^\circ)\\ \theta_{\text{degrees}} = \theta_{\text{radians}} &\cdot \frac{180^\circ}{\pi}\end{aligned} \tag{1}$$

Where $\lambda_1 = \frac{x_1 \cdot \pi}{180^\circ}$, $\lambda_2 = \frac{x_2 \cdot \pi}{180^\circ}$, $\varphi_1 = \frac{y_1 \cdot \pi}{180^\circ}$ and $\varphi_2 = \frac{y_2 \cdot \pi}{180^\circ}$. Bearings θ_{degrees} fall into different ranges, each with a corresponding cardinal direction (e.g., 'North-West'). (ii) Egocentric relations between landmarks and end point to the path, e.g., 'on the right side'. To calculate egocentric relations, we rely on two key angles (Eq. 1): the bearing of the path itself $\theta^p_{\text{degrees}}$, and the bearing of the shortest imaginary line connecting the path to the landmark's point $\theta^l_{\text{degrees}}$:

$$\Delta\theta^{l-p} = (\theta^l_{\text{degrees}} - \theta^p_{\text{degrees}}) \mod 360^\circ \tag{2}$$

$$\begin{cases}\text{'RIGHT'}, & \text{if } \Delta\theta^{l-p} < 180^\circ\\ \text{'LEFT'}, & \text{otherwise}\end{cases}$$

'LEFT' and 'RIGHT' indicate the landmark's position relative to the path. (iii) The number of blocks and intersections the agent must pass through to reach the end point. (iv) the end point's egocentric and allocentric position on the block, e.g., 'middle of the block' and 'north-east corner of the block'.

[2]Wikidata is Wikipedia's free, open, and interconnected knowledge base. https://www.wikidata.org/

Allocentric position on the block involves determining the bearing (Eq. 1) of the path along the block and mapping it to cardinal direction as in (i).

3.3 Data Generation

Based on the sampling and spatial relations' calculations we use two methods to generate the instruction: via templates created with a CFG (Chomsky, 1956), and via prompting an LLM — for enlarging the vocabulary and the style of the text.

CFG-based Method Here the key idea is using a CFG to generate templates that can then be adapted according to the sampled data. The CFG we design requires defining terminal symbols (lexical elements), nonterminal symbols, and production rules. The nonterminals contain the main parts of what a path description contains, such as descriptions around the goal, along the path, what to avoid, and so on. The terminals contain optional variations, for example, the verb for the agent to proceed can be 'go', 'walk', and so on. The grammar creates templates that are processed into instructions, as demonstrated in Figure 2. For a given sampled path, we randomly pick a template that contains all the landmark categories and spatial relations calculated for the path. We then generate the instruction by replacing the variables in the chosen template with the corresponding landmarks and spatial relations (e.g., '*NUMBER_INTERSECTIONS*' will be replaced with the actual number of intersections the agent should walk).

Prompting LLMs Using the aforementioned template-based instructions, we prompted an LLM to 'rephrase the subsequent navigation instruction, ensuring it explains how to travel from the starting position to the destination: *Navigation Instruction*', where the *Navigation Instruction* is an instruction based on the CFG generation process. For example, based on the example in Figures 1 and 2 we get the following sentence: 'Head northwest from St. Vincent de Paul Church for 2 intersections. The garden is next to a fast-food restaurant.

4 Experimental Setup

4.1 Evaluation

We follow the RVS evaluation metrics: (i) 100m accuracy; (ii) 250m accuracy for coarse-grained evaluation; (iii) mean absolute error distance (MAE); (iv) median absolute error (Med.AE); (v) maximum absolute error (Max.AE); and (vi) area under the curve (AUC) of the error distance. Here are the formulas for evaluating set S with metrics (iii-vi):

	RVS	Aug-CFG	Aug-Prompt	Aug-WikiGeo
Avg. Text Length	43.47	33.70	37.26	24.82
Avg.Entities	3.98	4.01	4.00	2.20

Table 1: Statistics over RVS, and augmentation data.

$$\text{MAE(S)} = \frac{1}{|S|}\sum_{s\in S} dist(loc(s), approx(s)) \quad (3)$$

$$\text{Med.AE(S)} = \{dist(loc(s), approx(s))|s \in S\}_{\lfloor |S|/2 \rfloor} \quad (4)$$

$$\text{Max.AE(S)} = max(\{dist(loc(s), approx(s))|s \in S\}) \quad (5)$$

$$\text{AUC(S)} = \frac{\int_0^{\infty} (\log dist(loc(s), approx(s)) + \epsilon | s \in S) \uparrow ds}{\log H_{max} \cdot (|S|-1)} \quad (6)$$

Where $\epsilon = 1e-5$ and $H_{max} = 20,037 \cdot 10^3$, approximately the maximum haversine distance.

4.2 Models for evaluation

T5-model We test our augmentation method with T5 model, a transformer-based encoder-decoder model designed with a text-to-text format (Raffel et al., 2020). Both encoder and decoder utilize multi-head, multi-layer self-attention mechanisms (Vaswani et al., 2017). Given input sequence text $X = (x_1, ..., x_N)$ and a starting point p_s, the encoder encodes the instruction and the starting point's representation such that $E^l = (e_1^l, ..., e_N^l, e_{p_s}^l)$ where $l \in L$, representing the L hierarchical encoded layers. The output of the final encoder layer is a sequence of hidden vectors $H = (h_1, ..., h_N, h_{p_s})$. The decoder generates output tokens sequentially, predicting the probability $p(p_t|p_{1:t-1}, H) = \text{softmax}(W_o \otimes h_t')$ of token p_t at step t, based on the previous outputs and hidden state h_t'. Importantly, the model is trained with a pre-defined high-level path P that guides the generation process. This path starts at the starting point, traverses through prominent landmarks ordered by their direction relative to the goal, and eventually reaches the goal itself.

Non-learning LANDMARK Baseline Predicts the location of a prominent landmark (defined in Sec. 3.2) in the map within a radius of 1 kilometer.

4.3 Data

RVS The RVS (Paz-Argaman et al., 2024) dataset serves as a human-level benchmark for the purpose of evaluating the ability to follow

	Method	Training Set	100m Accuracy	250m Accuracy	MAE	Med.AE	Max.AE	AUC
	Manhattan (Manh) Seen-city Development Results							
1	HUMAN	NA	88.12	95.64	74	4	2,996	0.10
2	LANDMARK	NA	0.54	5.26	776	815	1,384	0.39
3	T5	RVS Train-set	27.92 (0.39)	52.63 (0.45)	362 (9)	231 (3)	2,957 (641)	0.32 (0.00)
4	T5	Aug-WikiGeo Manh	0.00 (0.00)	1.54 (0.00)	1,085 (0)	1,124 (0)	1,929 (0)	0.41 (0.00)
5	T5	Aug-CFG Manh	28.83 (0.63)	46.15 (0.77)	668 (17)	304 (27)	4,637 (2,207)	0.34 (0.00)
6	T5	Aug-Prompt Manh	21.32 (0.20)	37.01 (0.14)	963 (17)	658 (14)	6,731 (1,003)	0.36 (0.00)
7	T5	Aug-CFG Manh & RVS Train-set	**45.97 (1.34)**	**64.01 (0.89)**	377 (32)	**121 (15)**	5,317 (831)	0.3 (0.00)
	Pittsburgh (Pitt) Unseen-Development Results							
8	HUMAN	NA	86.94	92.94	99	7	2,951	0.13
9	LANDMARK	NA	1.47	9.48	677	691	1,345	0.38
10	T5	RVS Train-set	0.49 (1.47)	2.34 (1.44)	1,171 (24)	1,107 (14)	4,701 (101)	0.41 (0.00)
11	T5	Aug-WikiGeo Pitt	0.00 (0.00)	2.05 (0.00)	961 (0)	955 (0)	1,912 (0.00)	0.40 (0.00)
12	T5	Aug-CFG Pitt	**46.63** (0.54)	**63.73** (0.41)	466 (5)	120 (1)	5,251 (0.00)	0.31 (0.00)
13	T5	Aug-Prompt Pitt	37.10 (0.49)	58.30 (0.34)	492 (5)	159 (1)	5,251 (0)	0.32 (0.00)
14	T5	RVS Train-set & Aug-CFG Pitt	46.24 (0.30)	62.85 (0.41)	**387** (17)	**116** (4)	5,162 (103)	0.30 (0.00)
	Philadelphia (Phila) Unseen-city Zero-shot Results							
15	HUMAN	NA	93.64	97.97	27	3	2,708	0.05
16	LANDMARK	NA	1.02	7.90	707	713	1,384	0.38
17	T5	RVS Train-set	0.26 (0.05)	1.80 (0.27)	1,362 (43)	1,308 (35)	6,911 (454)	0.42 (0.00)
18	T5	RVS Train-set & Aug-CFG Phila	**46.09 (0.50)**	**61.66 (0.00)**	**579 (2)**	**125 (3)**	5,774 (715)	0.31 (0.01)

Table 2: Results are divided over RVS's test (Philadelphia) and development sets (Manhattan and Pittsburgh). The distance errors are presented in meters. For the learning models, we report the mean over three random initializations, and the standard deviation (STD) is in brackets.

allocentric navigation instructions based on a map. It consists of English navigation directives, each paired with a start and end point. The data is divided into four distinct sets: (i) *Training-set* – containing 7,000 instructions from Manhattan; (ii) *Seen-city development-set* – containing 1,103 instructions from Manhattan; (iii) *Unseen-city development-set* – containing 1,023 instructions from Pittsburgh; (iv) *Test-set* – containing 1,278 instructions from Philadelphia.

The following datasets are all synthetically generated. We generated 200,000 instructions per dataset for each region, with the exception of Aug-WikiGeo:

Aug-CFG Data created with the CFG method described in Section 3. The CFG method created 194,721 templates with 15 production rules. The vocabulary contains only unique 111 tokens. Table 1 shows that this method produces shorter instructions than the RVS, but with more entities in each instruction. Figures 1 and 2 show an example of an instruction generated based on this method.

Aug-CFG-Allocentric Data created with a CFG including templates with allocentric spatial relations between entities. E.g., 'the school is north of the bar'. The data contains 69,720 templates.

Aug-CFG-Egocentric Data created using a CFG which contains templates with egocentric spatial relations between entities, e.g., 'the bar on your right'. The data contains 112,640 templates.

Aug-CFG-Minimal This dataset leverages a minimal set of templates (64) to represent the full range of entities and spatial relationships found in the Aug-CFG data. We constructed Aug-CFG-Minimal using a greedy selection process. We began by selecting the first template. Subsequently, we added templates that introduced new spatial features not covered by the existing set. This process continued until we reached a final set of 64 templates that comprehensively capture all possible spatial features.

Aug-Prompt Data generated by prompting PaLM2 (Anil et al., 2023) with the prompting method described in Section 3.[3] Table 1 shows that this method produces longer instructions than the CFG-based method but with fewer entities.

Aug-WikiGeo To facilitate a comparison with standard open-source location augmentation methods, such as Wikipedia-based (Krause and Cohen, 2023), we generated the comprehensive Wikigeo dataset by consolidating data from Wikipedia (pages and backlinks), Wikidata (pages), and OpenStreetMap (entities). The data for Manhattan (Manh., 255,663 samples), Pittsburgh (Pitt., 27,401 samples), and Philadelphia (Phila., 52,367 samples) are generated based on the same regions as in the evaluation. Table 1 shows that this method produces shorter instructions and fewer entities than the CFG-based and prompting-based methods.

[3] We use PaLM2 'models/text-bison-001' - for more details, see https://developers.generativeai.google

	Method	Training Set	100m Accuracy	250m Accuracy	MAE	Med.AE	Max.AE	AUC
			Manhattan (Manh) Seen-city Development Results					
1	**T5**	Aug-CFG Manh	28.83 (0.63)	46.15 (0.77)	668 (17)	304 (27)	4,637 (2,207)	0.34 (0.00)
2	**T5**	Aug-CFG-Allocentric Manh	29.83 (1.22)	47.96 (2.95)	681 (39)	310 (66)	6,446 (88)	0.34 (0.01)
3	**T5**	Aug-CFG-Egocentric Manh	28.83 (0.38)	45.87 (0.13)	751 (29)	384 (5)	4,572 (77)	0.35 (0.00)
4	**T5**	Aug-CFG-Minimal Manh	23.21 (0.39)	39.62 (0.13)	880 (12)	567 (47)	5,463 (2,228)	0.36 (0.00)
5	**T5**	Aug-Dummy Manh	1.35 (0.13)	5.62 (0.13)	1,240 (28)	1,130 (41)	4,654 (0.00)	0.41 (0.00)
			Pittsburgh (Pitt) Unseen-Development Results					
6	**T5**	Aug-CFG Pitt	**46.63** (0.54)	**63.73** (0.41)	466 (5)	120 (1)	5,251 (0.00)	0.31 (0.00)
7	**T5**	Aug-CFG-Allocentric Pitt	45.94 (0.48)	63.64 (0.55)	453 (18)	119 (0)	4,835 (91)	0.31 (0.00)
8	**T5**	Aug-CFG-Egocentric Pitt	43.01 (0.48)	60.22 (1.60)	514 (2)	131 (1)	6,034 (399)	0.31 (0.00)
9	**T5**	Aug-CFG-Minimal Pitt	42.82 (2.07)	62.84 (0.20)	479 (26)	132 (3)	3,160 (1030)	0.31 (0.00)
10	**T5**	Aug-Dummy Pitt	6.26 (0.83)	13.88 (1.04)	1,069 (6)	933 (1)	3,197 (746)	0.39 (0.00)

Table 3: CFG-based Augmentation Ablation Results

Aug-Dummy To assess the contribution of the text augmentation, as opposed to just learning all possible paths from a known starting point, we use an augmentation where the text does not convey any spatial details of the location. To create the non-spatial text, we used PaLM2 by requesting it to rephrase the sentence "Meet me here" and got a total of 31 versions. Path sampling followed the method described in Section 3.1.

5 Results

5.1 Analysis of Quantitative Results

Table 2 shows the results of our experiments over RVS's seen-city development set (Manh.), unseen-city development set (Pitt), and the unseen-city test set (Phila.). For the seen environment (Manh.), training on synthetic data only (line 5) outperforms human-annotated data (line 3). The gap is small, but if the model is first trained on synthetic data and then on human data (line 7), the gap is a 65% ratio in 100m accuracy and a 110m lower in Med.AE.

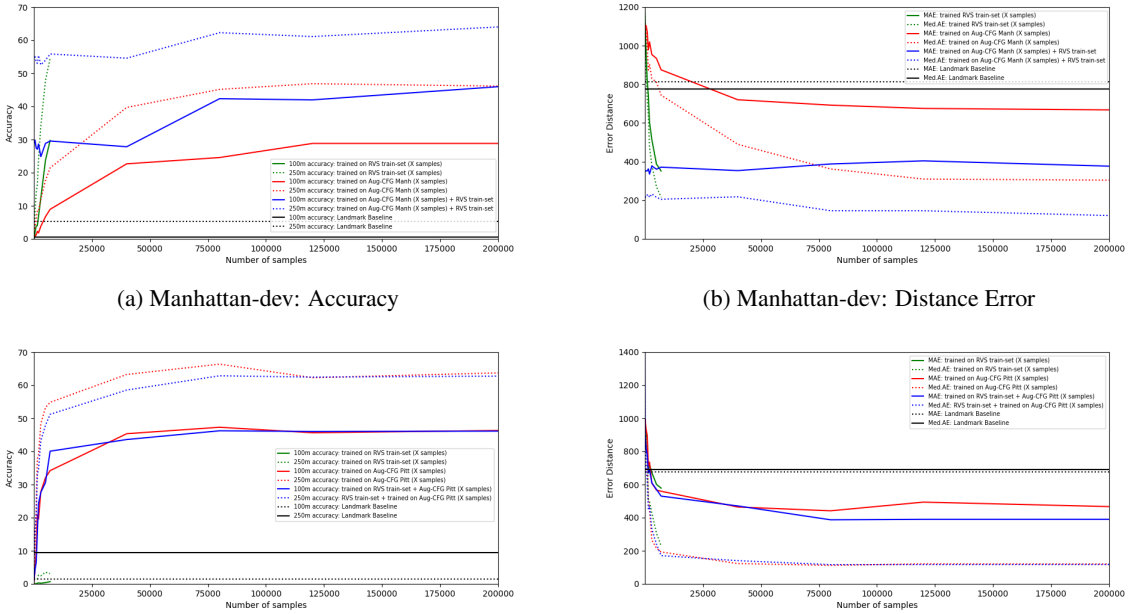
Training on real human data from other regions (lines 10, 17) fails to translate to unseen environments like Pittsburgh and Philadelphia. However, injecting region-specific synthetic data (line 12) dramatically boosts performance: 46.14% higher 100m accuracy and 987m lower Med.AE. WikiGeo's emphasis on local features sacrifices spatial relational understanding, leading to lower performance on all development sets (lines 4, 11). However, its superior max distance estimation (up to 2km) and lower distance error over models trained on human-annotated data from a different region (lines 11 vs. 10), suggest a strong ability to learn path boundaries based on localized information.

Aug-Prompt, while generating instructions with richer language and stylistic diversity compared to Aug-CFG's template-based approach, exhibits a marked decrease in performance (lines 6, 13 vs. lines 5, 12). Sampling 20 instructions from Aug-Prompt, we found that in two cases, the type of goal was emitted, and in five cases, the model 'hallucinated' – adding incorrect spatial relations. This indicates a potential trade-off between linguistic complexity and the fidelity of spatial information within generated instructions.

Table 3 examines variations of CFG-based augmentation. Consistent with the emphasis on allocentric spatial relations in RVS, training on Aug-CFG-Allocentric (lines 2, 7) surpasses Aug-CFG-Egocentric (lines 3, 8) in both development sets. However, results are mixed regarding whether training on Aug-CFG-Allocentric is better than training on Aug-CFG. In the seen environment, Aug-CFG-Allocentric (line 2) outperforms Aug-CFG (line 1) in accuracy but underperforms in error distance. In the unseen-environment (Pittsburgh), the opposite is true – Aug-CFG-Allocentric (line 7) underperforms Aug-CFG (line 6) in accuracy and overperforms error distance. This inconclusive evidence suggests potential value in investigating a single data approach for tasks with varying demands, such as RUN's reliance on egocentric relations.

Despite capturing identical spatial relations, Aug-CFG-Minimal's limited stylistic variations and vocabulary size due to fewer templates hinder its performance compared to Aug-CFG across both seen (line 1 vs. line 4) and unseen (line 6 vs. line 9) sets. This suggests that the mere presence of accurate spatial relations may not be sufficient for optimal model learning, potentially due to insufficient exposure to diverse linguistic contexts and syntactic structures.

Training on Aug-Dummy teaches the model only the optional paths from a starting point, as the instruction is a dummy one. The poor results achieved over Aug-Dummy in both seen and unseen environments (lines 5, 10), prove that training on Aug-CFG allows the model to learn spatial relations from the instructions. The Aug-Dummy for



(a) Manhattan-dev: Accuracy

(b) Manhattan-dev: Distance Error

(c) Pittsburgh-dev: Accuracy

(d) Pittsburgh-dev: Distance Error

Figure 3: T5 performance (Y-axis) with varying AUG-CFG training samples (X-axis). (a,b) RVS seen-city (Manhattan), (c,d) RVS unseen-city (Pittsburgh).

Pittsburgh results are better as Manhattan is much denser in entities than Pittsburgh, thus, it contains more optional paths from the starting point.

5.2 Data Quantity Impact

Figure 3 shows four graphs of T5 performance trained on different amounts of AUG-CFG data. The results reveal a quality-quantity trade-off influencing T5's performance. In seen cities (a, b), 7,000 high-quality human annotations (green lines) outperform 7,000 synthetic AUG-CFG samples (red lines) in 100m accuracy (27.92% vs. 8.93%) and Med.AE (231m vs. 744m), indicating that the human-annotated RVS data possesses substantially higher quality than the synthetic data. The steep curve of the RVS train-set (green lines) compared to the mild curve of AUG-CFG (red lines) further reinforces this conclusion. However, this trend flips with 200K synthetic samples, showcasing the power of quantity over quality when data is abundant. This suggests ample data helps the model grasp the environment and spatial relations. Additionally, while both RVS train-set (green line) and AUG-CFG (red line) demonstrate strong performance, the combined AUG-CFG + RVS train-set (blue line) exhibits a sustained upward trend, consistently surpassing the green and red lines in both accuracy and distance error. This further indicates that augmenting with a large amount of data can potentially enhance performance beyond even high-quality human annotations. This approach offers a promising solution for the data scarcity challenges often encountered in NLP geospatial tasks.

In the unseen-city split (c, d), the RVS train-set (green line) exhibits steady improvements in accuracy and error distance despite originating from a different region. This demonstrates that high-quality data, even when geographically distinct, can benefit the model. However, the significant performance gap between the RVS train-set (green) and AUG-CFG (red) lines, even with equal data quantities, reaffirms the importance of regionally-specific data. While the RVS train-set (green) slope nears AUG-CFG (red line), its accuracy improvement remains significantly lagged (moderate slope), suggesting the model primarily learns general directional understanding rather than fine-grained spatial reasoning. Furthermore, the combined AUG-CFG + RVS train-set (blue) offers only marginal gains over AUG-CFG alone (red), indicating that the benefits of high-quality non-regional data diminish when paired with large-scale augmentation.

5.3 Distribution Analysis

Figure 4 reports the performance of the various augmentation methods through cumulative distribution functions (CDFs). These CDFs show the percentage of inferences with error distances below

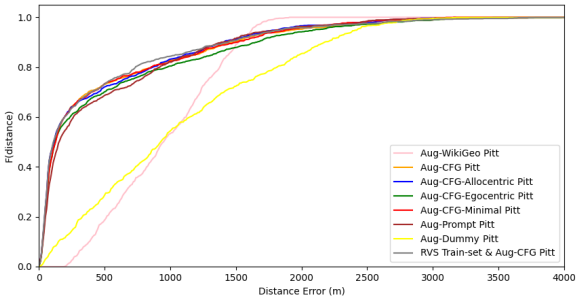


Figure 4: Cumulative distribution function (CDF) error. Augmentation impact on distance error (meters).

a specific meter value (x-axis), effectively capturing accuracy across all error values and exposing underlying error distributions. Notably, CFG-based methods (AUG-CFG and its variants) exhibit a remarkably similar distribution, characterized by a sharp accuracy ascent up to approximately 200m. Aug-Dummy's curve (yellow) resembles a ray emanating from the origin, suggesting a strong correlation with the path distribution assimilated during training. Aug-WikiGeo's low-resolution nature is evident in its CDF (pink). While it starts slow ($\tilde{2}$00m) and climbs faster than the yellow line, its limited accuracy holds it back. It only catches up to the CFG-based methods at $\tilde{1}$500m, but then fizzles out (plateaus) before they even reach their peak.

6 Background and Related Work

Text-based navigation tasks constitute a multimodal challenge (Antol et al., 2015; Paz-Argaman et al., 2020; Ji et al., 2022) demanding the integration of language comprehension and environmental knowledge. This environment can be either indoor, commonly represented through the agent's visual perception as it navigates (Anderson et al., 2018), or outdoor. For outdoor environments, representation falls into three main categories: (i) *Visual Representation* — similar to indoor settings, real-world imagery can be employed, with agents learning the environment through street-view like exploration (Anderson et al., 2018); (ii) *Map-based Representation* — agents navigate based on map perception, as seen in studies utilizing maps as the primary input (Anderson et al., 1991; Paz-Argaman et al., 2024); and (iii) *Hybrid Representation* — a combination of visual and map information (Vasudevan et al., 2020; de Vries et al., 2018).

Realistically, we would like to learn to navigate based on text in new environments, ones that our models did not train on before. Furthermore, environments constantly change, such that we need to reacquaint our models with these changes in order for them to stay relevant (Zhang and Choi, 2021).

Synthetic data generation, facilitated by LLMs, has become a prominent approach in NLP to mitigate data (Sahu et al., 2022; Stylianou et al., 2023). This technique is particularly attractive due to its adaptability to various tasks and its ability to generate substantial volumes of data, allowing for robust model training and improved performance. Data generation has also been applied to multimodal tasks, demonstrating performance close to that of human-annotated data (Bitton et al., 2021).

Previous works on vision and language navigation tasks (VLN) tried to tackle the data scarcity issue by generating synthetic data with a generative model trained with human annotated data (Fried et al., 2018; Zhu et al., 2020; Majumdar et al., 2020). However, the learned distribution was limited to the environment the model was trained on. EnvEdit (Li et al., 2022b) tackled the VLN task by generating new environments and synthetic navigation instructions in order to teach models to generalize to new environments. The new environments created differ in style, appearance, or the configuration of the objects. However, a lack of new objects hinders unseen object handling. Kamath et al. (2023) leveraged image-to-image synthesis via a Generative Adversarial Network (GAN) architecture (Koh et al., 2023) to create new viewpoints for existing environments and subsequently generate synthetic instructions for these environments.

Several prior works have tackled unseen environments without employing data generation: (i) *Entity abstraction* – Paz-Argaman and Tsarfaty (2019) propose learning spatial language independent of specific entities by linking abstracted entities to the agent's perception. However, this approach is limited to simple navigation tasks where the agent has limited perception (i.e., line of sight). It struggles with complex tasks like RVS, which require allocentric spatial reasoning. (ii) *cross-lingual augmentation* – Li et al. (2022a) leverage spatial data across different languages for augmentation. However, it still requires some environment-specific annotations, even if not in the target language. Additionally, geolocation tasks utilizing open-source datasets like Wikipedia often rely on grid-based representations, suffering from spatially unoriented descriptions. This leads to significant retrieval er-

rors exceeding tens of kilometers, limiting their accuracy (Wing and Baldridge, 2011). (iii) *Graph-based representation* – representing the environment via a graph and learning the connection between environment and language (Paz-Argaman et al., 2023, 2024). This approach lacks promising results due to challenges in encoding complex spatial relationships within a graph format. Schumann and Riezler (2021) trained a neural network to generate synthetic navigation instructions based on OpenStreetMap representations. However, these instructions were limited to step-by-step, local line-of-sight guidance. Furthermore, the model's performance was evaluated on unseen paths within the same city it was trained on.

This work tackles the problem of generating synthetic instructions for unseen environments, exemplified by the RVS dataset (Paz-Argaman et al., 2024). We propose two techniques: an LLM-based approach for its adaptability, and a CFG rule-based approach for increased precision, highlighting the trade-off between data-driven efficiency and human-crafted accuracy.

7 Conclusion

This work presents a novel data-augmentation solution for spatial NLP tasks. Leveraging spatial relation extraction it generates a vast, albeit slightly less refined, dataset compared to human annotations. This quantitative advantage unlocks superior performance, as evidenced by a 44.54% absolute improvement in 100m accuracy and a 1,170m reduction in median absolute error distance on unseen environments in the RVS dataset. These results demonstrate the effectiveness of our approach in handling novel environments with sparse or no human data. Moreover, our CFG-based augmentation method offers adherence to correct spatial relations, control over the content, and interpretability, over LLMs in spatial descriptions and could serve as a future tool for evaluation, detection, and mitigation of artifacts such as 'hallucinations'.

Limitations

Data Dependence This paper presents a promising data augmentation method for enhancing NLP tasks like outdoor geolocation and navigation. However, its effectiveness hinges on accessing comprehensive geospatial data, which can be difficult to find in indoor (Anderson et al., 2018; Jain et al., 2019; Nguyen et al., 2019; Thomason et al., 2020; Qi et al., 2020; Ku et al., 2020) and virtual environments (MacMahon et al., 2006; Yan et al., 2018; Misra et al., 2018; Shridhar et al., 2020; Kim et al., 2020). Furthermore, the reliance on open-source data introduces the risk of incompleteness or regional unavailability.

Rule-based Scalability vs. LLM-Generated Artifacts In addition, the leading augmentation method (CFG-based) used in this study is rule-based, which, while offering control, preciseness, and interpretability, also requires careful rule design. Crafting effective rules can be time-consuming, laborious, and require substantial expertise in the specific domain. This complexity can limit the scalability and adaptability of the method to new situations or contexts. Furthermore, encoding all necessary knowledge into explicit rules can be challenging, especially for complex domains. This can limit the method's ability to capture subtle nuances or unforeseen situations. This approach also stands in contrast to the prevailing augmentation methods, which are grounded in LLMs (Schick and Schütze, 2021; Wang et al., 2022). Using LLMs to generate data for augmentation represents a promising avenue but requires further research to address issues like hallucinations, which are particularly critical in spatial descriptions. Therefore, rule-based methods such as our CFG-based method, which are not vulnerable to hallucination problems, could serve as a valuable tool for future research exploring the evaluation, detection, and mitigation of such artifacts in LLM-generated text. Furthermore, by comparing augmentations generated using both methods on specific tasks, we might gain insights into the types of artifacts introduced by LLMs and how rule-based methods can help mitigate them.

Limited Perception: Lack of Visual Cues Our study primarily relied on map-based knowledge for navigation tasks, which deviates from how humans navigate in natural settings. Real-world navigation often involves integrating both visual cues and spatial knowledge acquired from maps, a complexity not fully captured in our current approach. While the StreetLearn dataset (Mirowski et al., 2019) offers Google Street View imagery for the test environments over both the Manhattan and Pittsburgh regions in the RVS setup, we did not leverage this visual information. Future research could extend the scope of our study by integrating visual perception and map-based knowledge into the text

generation approach for augmentation.

Bridging the Human-AI Performance Gap In spite of progress made in this research, a substantial gulf still separates current models' performance from human performance in the RVS task. Even with our augmentation methods, current models lag behind human performance by 47.55% and 42.15% in 100-meter accuracy for seen and unseen environments, respectively. Bridging this gap presents a critical challenge and an exciting opportunity for future research, potentially unlocking novel avenues for pushing the boundaries of this task.

Acknowledgements

This research has been funded by the European Research Council (ERC), grant number 677352 and by a grant from the Israeli Science Foundation (ISF) number 670/23, for which we are grateful. The research was further supported by a KAMIN grant from the Israeli Innovation Authority, and computing resources kindly funded by a VATAT grant and via the Data Science Institute from Bar-Ilan University (BIU-DSI). We are also grateful for the additional support provided by a Google grant.

References

Kwasi Abebrese. 2019. *Implementing street addressing system in an evolving urban center. A case study of the Kumasi metropolitan area in Ghana*. Ph.D. thesis, Iowa State University.

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 (CVPR)*, pages 3674–3683.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Yonatan Bitton, Gabriel Stanovsky, Roy Schwartz, and Michael Elhadad. 2021. Automatic generation of contrast sets from scene graphs: Probing the compositional consistency of gqa. *arXiv preprint arXiv:2103.09591*.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

Aleksandra Edwards, Asahi Ushio, Jose Camacho-Collados, Hélène de Ribaupierre, and Alun Preece. 2021. Guiding generative language models for data augmentation in few-shot text classification. *arXiv preprint arXiv:2111.09064*.

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. *Advances in Neural Information Processing Systems*, 31.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Yingjie Hu, Gengchen Mai, Chris Cundy, Kristy Choi, Ni Lao, Wei Liu, Gaurish Lakhanpal, Ryan Zhenqi Zhou, and Kenneth Joseph. 2023. Geo-knowledge-guided gpt models improve the extraction of location descriptions from disaster-related social media messages. *International Journal of Geographical Information Science*, pages 1–30.

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. *arXiv preprint arXiv:1905.12255*.

Anya Ji, Noriyuki Kojima, Noah Rush, Alane Suhr, Wai Keen Vong, Robert D Hawkins, and Yoav Artzi. 2022. Abstract visual reasoning with tangram shapes. *arXiv preprint arXiv:2211.16492*.

Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. 2023. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10813–10823.

Hyounghun Kim, Abhay Zala, Graham Burri, Hao Tan, and Mohit Bansal. 2020. Arramon: A joint navigation-assembly instruction interpretation task in dynamic environments. *arXiv preprint arXiv:2011.07660*.

Jing Yu Koh, Harsh Agrawal, Dhruv Batra, Richard Tucker, Austin Waters, Honglak Lee, Yinfei Yang,

Jason Baldridge, and Peter Anderson. 2023. Simple and effective synthesis of indoor 3d scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1169–1178.

Amir Krause and Sara Cohen. 2023. Geographic information retrieval using wikipedia articles. In *Proceedings of the ACM Web Conference 2023*, pages 3331–3341.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Jialu Li, Hao Tan, and Mohit Bansal. 2022a. Clear: Improving vision-language navigation with cross-lingual, environment-agnostic representations. *arXiv preprint arXiv:2207.02185*.

Jialu Li, Hao Tan, and Mohit Bansal. 2022b. Envedit: Environment editing for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15407–15417.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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

Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 259–274. Springer.

Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, Denis Teplyashin, Karl Moritz Hermann, Mateusz Malinowski, Matthew Koichi Grimes, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, et al. 2019. The streetlearn environment and dataset. *arXiv preprint arXiv:1903.01292*.

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. 2019. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12527–12537.

Tzuf Paz-Argaman, Tal Bauman, Itai Mondshine, Itzhak Omer, Sagi Dalyot, and Reut Tsarfaty. 2023. Hegel: A novel dataset for geo-location from hebrew text. *arXiv preprint arXiv:2307.00509*.

Tzuf Paz-Argaman, Sayali Kulkarni, John Palowitch, Reut Tsarfaty, and Jason Baldridge. 2024. Where do we go from here? multi-scale allocentric relational inferencefrom natural spatial descriptions. In *EACL2024*. Association for Computational Linguistics.

Tzuf Paz-Argaman and Reut Tsarfaty. 2019. RUN through the streets: A new dataset and baseline models for realistic urban navigation. 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 6449–6455, Hong Kong, China. Association for Computational Linguistics.

Tzuf Paz-Argaman, Reut Tsarfaty, Gal Chechik, and Yuval Atzmon. 2020. ZEST: Zero-shot learning from text descriptions using textual similarity and visual summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 569–579, Online. Association for Computational Linguistics.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.

Mark Sanderson and Janet Kohler. 2004. Analyzing geographic queries. In *SIGIR workshop on geographic information retrieval*, volume 2, pages 8–10.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540*.

Raphael Schumann and Stefan Riezler. 2021. Generating landmark navigation instructions from maps as a graph-to-text problem. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 489–502, Online. Association for Computational Linguistics.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Yuval Solaz and Vitaly Shalumov. 2023. Transformer based geocoding. *arXiv preprint arXiv:2301.01170*.

Amanda Spink, Bernard J Jansen, Dietmar Wolfram, and Tefko Saracevic. 2002. From e-sex to e-commerce: Web search changes. *Computer*, 35(3):107–109.

Nikolaos Stylianou, Despoina Chatzakou, Theodora Tsikrika, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2023. Domain-aligned data augmentation for low-resource and imbalanced text classification. In *European Conference on Information Retrieval*, pages 172–187. Springer.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.

UPU. 2012. *Addressing the world – An address for everyone*.

Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. 2020. Talk2nav: Long-range vision-and-language navigation with dual attention and spatial memory. *International Journal of Computer Vision*, pages 1–21.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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*.

Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. 2022. Less is more: Generating grounded navigation instructions from landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15428–15438.

Benjamin Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 955–964.

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*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 2021. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*.

Michael JQ Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. *arXiv preprint arXiv:2109.06157*.

Wanrong Zhu, Xin Eric Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, and William Yang Wang. 2020. Multimodal text style transfer for outdoor vision-and-language navigation. *arXiv preprint arXiv:2007.00229*.

A Data Generation

Our CFG rules define navigation instructions by combining five key elements in various orders:

- **Goal Description:** This specifies the target location.
- **Main Path:** It outlines the primary route using landmarks.
- **Approaching Goal:** This details how to get close to the target.
- **Goal Landmarks:** It describes landmarks relative to the final position.
- **Off-Path Awareness:** This identifies elements to avoid while navigating.

Table 4 demonstrates five navigation instructions and their corresponding CFG-templates.

B Models

We also test on the T5+GRAPH presented in RVS, which is a T5 model incorporating an OSM-based graph representation of the environment.

Exaples of CFG generated templates	Example of generated instruction from template
"Meet at the END_POINT. Go CARDINAL_DIRECTION from MAIN_PIVOT for INTERSECTIONS intersections. It will be near a NEAR_PIVOT. If you reach BEYOND_PIVOT, you have gone too far."	"Meet at the library. Go north-east from Starbucks for two intersections. It will be near a book store. If you reach a Zara cloth shop, you have gone too far."
"Go CARDINAL_DIRECTION from MAIN_PIVOT for BLOCKS blocks to arrive at the END_POINT, right GOAL_POSITION. You will pass MAIN_NEAR_PIVOT before reaching the destination. You've overshot the meeting point if you reach BEYOND_PIVOT."	"Go east from Grace Church for 3 blocks to arrive at the parking lot, right in the middle of the block. You will pass Jefferson Market Garden before reaching the destination. You've overshot the meeting point if you reach Chipotle."
"Go CARDINAL_DIRECTION from MAIN_PIVOT for BLOCKS blocks to arrive at the END_POINT, right GOAL_POSITION. You will see MAIN_NEAR_PIVOT before reaching the destination. You've overshot the meeting point if you reach BEYOND_PIVOT."	"Go east from Grace Church for 3 blocks to arrive at the parking lot, right in the middle of the block. You will see Jefferson Market Garden before reaching the destination. You've overshot the meeting point if you reach Chipotle."
"Walk CARDINAL_DIRECTION and past MAIN_PIVOT to reach the END_POINT. The END_POINT is not far from NEAR_PIVOT."	"Walk north and past Washington Square Park to reach the cafe. The cafe is not far from a tobacco shop."
"Head to MAIN_PIVOT and go CARDINAL_DIRECTION and meet at the END_POINT, right GOAL_POSITION."	"Head to Washington Square Park and go north and meet at the cafe, right on the southeast corner of the block."

Table 4: Examples of navigation instructions and the Context-Free Grammar (CFG)-derived templates they are created from.

B.1 The Graph Representation

A location can be represented by its position (*where* the location is) or by its semantics (*what* is present at the location, e.g., 'a bar'). Semantic knowledge is crucial for grounding mentioned entities to their physical references in the environment. To this end, we aim to represent the semantics via the RVS map-graph. We use the RVS map-graph and connect each node to its corresponding S2-cells. As the S2-geometry is a hierarchical structure, we allow for multiple levels of S2-cells connections. Also there are edges between neighboring S2-cells at a given level (see bottom part in Figure 5). To learn an embedding for each S2-cell in the environment, we compute random walks on the graph using node2vec algorithm (Grover and Leskovec, 2016). Following Yu et al. (2021), we use linear projection to cluster the graph embeddings into K categories using the k-means algorithm with cosine similarity distance. A new token is assigned to each category and added to the tokenizer's vocabulary. We perform multiple clusters and pass the graph's tokens with the instruction's tokens to the transformer encoder.

B.2 Experimental Setup Details

The Graph Embedding The graph was constructed using three levels of S2-Cells: 15, 16, and 17. At level 16, each sub-graph consisting of four neighboring S2-Cells was fully connected. All S2-Cells in the graph were linked to their parent S2-Cell based on the S2-geometry's hierarchy (i.e., level 17 S2-Cells were connected to level 16 S2-Cells and level 16 S2-Cells were connected to level 15 S2-Cells). Extracted entities from OSM and Wikidata were linked to the smallest level 17 S2-Cell that encompassed their geometry. The node of the entity included additional data such as their geometry, type and name of entity. Random walks on the graph were performed using node2vec (Grover and Leskovec, 2016).

For both T5-base models we use a pre-trained 'T5-Base' model from Hugging Face Hub, which is licensed under the Apache License 2.0. The T5 model was trained on the Colossal Clean Crawled Corpus (C4, Raffel et al. (2020)). The cross-entropy loss function was optimized with AdamW optimizer (Loshchilov and Hutter, 2017). The hyperparameter tuning is based on the average results run with three different seeds. We used a learning rate of 1e-4. The S2-cell level was searched in [15, 16, 17, 18] and 16 was chosen. The number of clusters for the quantization process was searched in [50, 100, 150, 200, 250] and 150 was chosen. We used 2 quantization layers. Number of epochs for early stopping was based on their average learning curve. We used the following parameters for the node2vec algorithm: an embedding size of 1024, a walk length of 20, 200 walks, a context window size of 10, a word batch of 4, and 5 epochs.

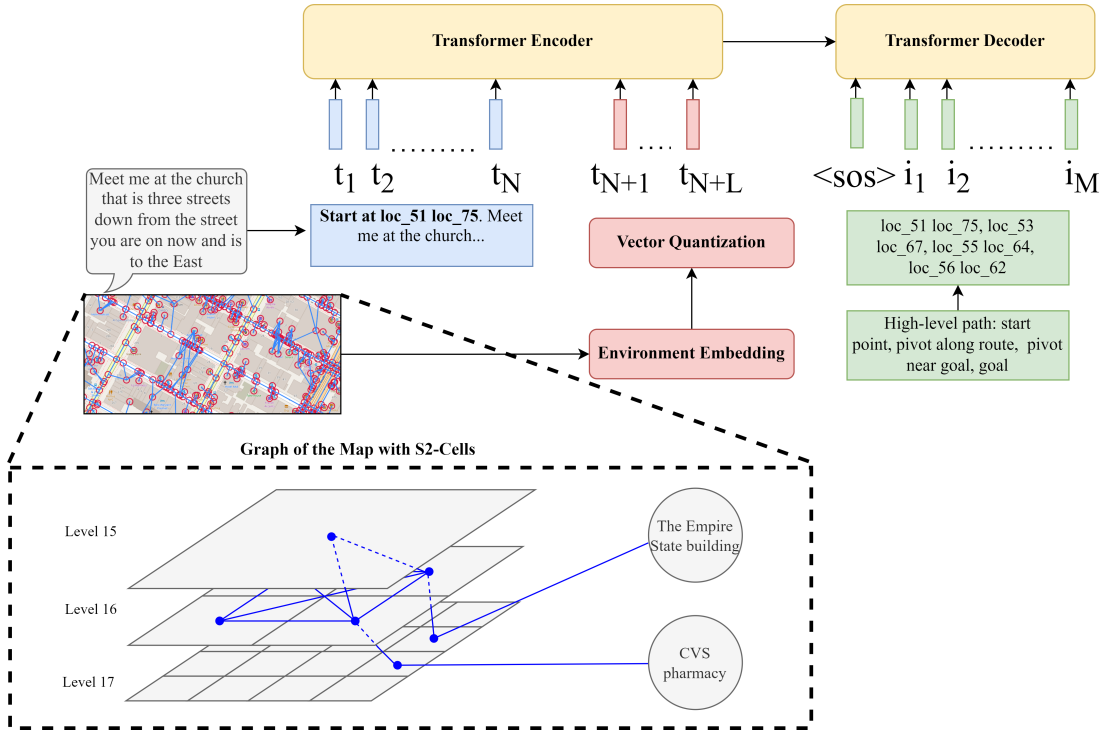


Figure 5: The RVS model based on a T5 transformer and a graph representation of the environment (Paz-Argaman et al., 2024).

C Additional Results

Table 5 demonstrates the performance of T5+GRAPH with CFG augmentation. The T5+GRAPH model's performance is improved by the CFG augmentation in both seen and unseen environments (lines 2 vs 4, and lines 8 vs 10). In the seen environment, T5+GRAPH with CFG augmentation achieves higher scores than T5 with CFG augmentation (lines 3 vs 4). However, there is no clear evidence to suggest which model performs better when using CFG augmentation data in both seen and unseen environments. This finding suggests that the explicit spatial knowledge offered by CFG augmentation data provides an easier path to learn spatial relations, making the graph information in T5+GRAPH redundant or even detrimental.

	Method	Training Set	100m Accuracy	250m Accuracy	MAE	Med.AE	Max.AE	AUC
	Manhattan (Manh) Seen-city Development Results							
1	**T5**	RVS Train-set	27.92 (0.39)	52.63 (0.45)	362 (9)	231 (3)	2,957 (641)	0.32 (0.00)
2	**T5+GRAPH**	RVS Train-set	29.40 (1.18)	54.67 (1.04)	**357** (7)	216 (8)	3,889 (826)	0.31 (0.01)
3	**T5**	Aug-CFG Manh	28.83 (0.63)	46.15 (0.77)	668 (17)	304 (27)	4,637 (2,207)	0.34 (0.00)
4	**T5+GRAPH**	Aug-CFG Manh	30.25 (0.95)	48.11 (0.92)	660 (24)	299 (17)	4,447 (677)	0.34 (0.01)
5	**T5**	Aug-CFG Manh & RVS Train-set	**45.97 (1.34)**	**64.01 (0.89)**	377 (32)	**121 (15)**	5,317 (831)	0.3 (0.00)
6	**T5+GRAPH**	Aug-CFG Manh & RVS Train-set	45.42 (0.9)	63.1 (1.41)	388 (10)	131 12)	3,162 (43)	0.3 (0.01)
	Pittsburgh (Pitt) Unseen-Development Results							
7	**T5**	RVS Train-set	0.49 (1.47)	2.34 (1.44)	1,171 (24)	1,107 (14)	4,701 (101)	0.41 (0.00)
8	**T5+GRAPH**	RVS Train-set	0.49 (1.01)	2.91 (1.37)	1,067 (77)	1,039 (56)	4,102 (727)	0.40 (0.00)
9	**T5**	Aug-CFG Pitt	46.63 (0.54)	63.73 (0.41)	466 (5)	120 (1)	5,251 (0)	0.31 (0.00)
10	**T5+GRAPH**	Aug-CFG Pitt	**46.67** (1.45)	63.1 (1.59)	474 (1)	119 (8)	5,251 (0)	0.31 (0.00)
11	**T5**	RVS Train-set & Aug-CFG Pitt	46.24 (0.30)	62.85 (0.41)	**387** (17)	**116** (4)	5,162 (103)	0.30 (0.00)
12	**T5+GRAPH**	RVS Train-set & Aug-CFG Pitt	45.75 (0.62)	**63.93** (0)	467 (7)	125 (1)	6,509 (755)	0.31 (0.00)

Table 5: T5+Graph Results: Results are divided over RVS's development sets. The augmentations data used for training depends on the method and region that corresponds to the evaluation region: Manhattan (Manh) and Pittsburgh (Pitt). The distance errors are presented in meters. For the learning models, we report the mean over three random initializations and the standard deviation (STD) is in brackets.