From Route Instructions to Landmark Graphs

Christopher Cervantes

HERE Technologies
425 West Randolph Street
Chicago, Illinois 60606
christopher.cervantes@here.com

Abstract

Landmarks are central to how people navigate, but most navigation technologies do not incorporate them into their representations. We propose the landmark graph generation task (creating landmark-based spatial representations from natural language) and introduce a fully end-to-end neural approach to generate these graphs. We evaluate our models on the SAIL route instruction dataset, as well as on a small set of real-world delivery instructions that we collected, and we show that our approach yields high quality results on both our task and the related robotic navigation task.

Introduction

As location technology has improved, there is an increased reliance on mobile and in-car apps to help people navigate in their daily lives. While these tools are well-suited to driving on established road networks, they rely on a precise geometric world representation that limits their usefulness in other navigation tasks (Zang et al. 2018).

When navigating, people use landmarks to orient themselves and define their surroundings rather than using coordinates and distance measures (Fellner, Huang, and Gartner 2017). Consequently, the techniques that are appropriate for automotive navigation aren't as useful when trying to find a side entrance when delivering a package, search through an unfamiliar area in an emergency, or locate a building in parts of the developing world where addressing is not well-defined. In all such cases, representing a route's landmarks relative to one another can be more useful than coordinate-based localization.

We propose a method for automatically extracting landmark and relation information from route instructions. Specifically, given a natural language route instruction (e.g. "Go away from the lamp to the intersection..."), our goal is to produce a *landmark graph* such as that shown in Figure 1, where nodes represent semantically meaningful locations (landmarks or decision points) and where edges indicate spatial information.

Route instructions allow people to communicate complex spatial information, which they do in part by focusing on
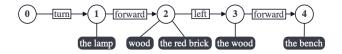
Figure 1: Landmark graph for: "Go away from the lamp to the intersection of the red brick and wood. Take a left onto the wood. Position one is one section down at the bench."

the salient landmarks needed to define and travel through an environment. When interpreted, however, route instructions require a person to correctly reconstruct the spatial information into a mental model. This can be challenging because the instruction writer and reader may talk about space differently, which can lead to divergent mental models. Our approach can help to bridge this gap. By extracting spatial information from instructions, we can create consistent, landmark-based spatial representations – landmark graphs – that can be useful in navigation tasks. Moreover, our approach's focus on extracting complete spatial representations may also be useful for similar tasks like robotic navigation, where spatial information consists of landmarks and traversal actions.

The main technical contribution of our approach is the joint prediction of landmark spans and actions from route instructions through the incorporation of an additional attention mechanism into an encoder-decoder model. This joint model yields significant improvements over the action-only baseline when evaluating action prediction performance. We also consider the introduction of the landmark graph generation task – that is, creating complete spatial representations from natural language – to be an important contribution, and show that our joint approach predicts landmark graphs with a high degree of similarity to the ground truth.

Related Work

In traditional geometric representations of space, a location is an abstraction independent from a referent. A latitude / longitude pair, for example, is meant to define a fixed point on the globe. While these representations can be useful, landmarks play a much more central role in navigation (Fellner, Huang, and Gartner 2017). In this context, a landmark

is a physical object [1] that can be used as a reference point in mental representations of space (Kai-Florian and Winter 2016). Seen in this way, landmarks are not *at* a location, but *define* a location. This is the core idea behind our location graphs: locations are defined not by coordinates but by their relationships to landmarks.

Similar landmark-based spatial representations have been proposed, from topological maps – graphs where nodes represent places and edges denote traversability or connection (Landsiedel et al. 2017) – to navigation graphs – where nodes represent landmarks or decision points and edges represent paths of travel (Yang and Worboys 2015). While these kinds of representations are often studied in the context of indoor navigation tools (Tsetsos et al. 2006; Yang and Worboys 2015), they are more useful generally, particularly as an intermediate step between natural language and the navigation task (Fellner, Huang, and Gartner 2017; Zang et al. 2018).

Natural language route instructions are a common way to communicate spatial information at pedestrian resolutions (e.g. "turn left at the fountain and continue down the hall"). Significant work has been done to understand and extract spatial meaning from these instructions in the robotic navigation domain (MacMahon, Stankiewicz, and Kuipers 2006; Kollar et al. 2010; Chen and Mooney 2011; Mei, Bansal, and Walter 2016; Duvallet et al. 2016; Chen et al. 2019; Anderson et al. 2018). In that setting, models are often trained to guide an autonomous agent through a virtual environment using unstructured language; the input to the system is a route instruction, the output a sequence of actions the agent must take, and the measure of success is whether the agent reached the goal destination.

This line of inquiry is similar to our own in that spatial information is extracted from natural language, but the goals differ. In the robotic navigation literature, guiding an autonomous agent can be thought of as a search problem; the world is represented as a grid or a graph, and the system must find a path from start to finish. In our setting, we want to represent a real-world environment by constructing a landmark graph. Robotic navigation can thus consider the spatial representation to be latent; as long as the agent reaches the goal, the model is successful. Our task, however, is concerned explicitly with this spatial representation.

In practice, this means that while robotic navigation approaches assume the agent will have access to spatial information at inference time through images (Chen et al. 2019; Anderson et al. 2018) or object labels (MacMahon, Stankiewicz, and Kuipers 2006; Chen and Mooney 2011; Mei, Bansal, and Walter 2016), our approach treats this information as part of the output.

Despite these differences, we borrow an important concept from the robotic navigation literature: the decomposition of the navigation task into *states* and *actions* (Chen and Mooney 2011; Mei, Bansal, and Walter 2016). In their framing, navigation is the process by which an agent takes actions (moving or turning) traversing from one decision point to another. Each decision point is predefined (e.g. grid intersections) and associated with a (possibly empty) set of world states describing nearby actions.

Informed by the landmark literature, our approach assumes no predefined decision points; they exist only in relation to nearby landmarks and the spatial relations (indicated by actions) to other decision points. This reframing allows us to adapt approaches from the robotic navigation literature to the task of generating landmark graphs.

We consider the graph's primary use to be as a navigational aid in real-world environments where a-priori knowledge about the space described in a route instruction is unavailable. Without this knowledge, however, our approach must identify which parts of the sentences refer to landmarks. Finding these *landmark spans* is similar to mention detection for coreference resolution (Peng, Chang, and Roth 2015; Lee et al. 2017) or referring expression detection for reference resolution (grounding) (Krishnamurthy and Kollar 2013; Kong et al. 2014; Kennington and Schlangen 2015; Plummer et al. 2015; Plummer et al. 2017). In such work, relevant noun phrases must be found as part of a larger task: clustering mentions or linking expressions to image referents. Similarly, our approach must identify landmark spans to define the space described by a route instruction.

By combing landmark spans with actions linking decision points, we develop the first fully end-to-end mechanism for generating landmark-based spatial representations from natural language route instructions.

Task

We consider the task of landmark graph generation from natural language route instructions: free-form imperative statements (single or multi-sentence) that can be used to guide an agent through an environment. Landmark graphs are composed of two types of nodes – decision points and landmarks – and two types of edges – between decision points and landmarks (indicating nearness) and directed from one decision point to another with an action label (indicating the path of traversal and thus the spatial relation between points).

The task can be considered a form of summarization by which action sequence $\mathbf{a}$ and world state (landmark) sequence $\mathbf{s}$ are extracted from instruction $\mathbf{w}$ ($w_i \in \mathbf{w}$). These same-length sequences ($|\mathbf{a}| = |\mathbf{s}|$) represent a path of traversal; each step (decision point) in the path has nearby landmarks (s_t) and an action indicating how the current step was reached (a_t).

We consider nine actions: *stand*, *forward*, *left*, *right*, *ascend*, *descend*, *turn*, *move*, and *STOP* ($a_t \in \{s, f, l, r, a, d, t, m, \langle STOP \rangle\}$). In addition to the *forward*, *right*, *left*, *stand*[2], and *STOP* actions that are standard in the literature, we also include *ascend* and *descend* to account for three dimensional movement (e.g. climbing stairs). We also include the ambiguous *turn* and

---

[1] Places defined by physical objects may also be landmarks; e.g. "the end of the hall" defines a component of an object.

[2] Our *stand* action, like `verify` in the SAIL dataset, is used as a representational tool associating landmarks with decision points (i.e. when a sentence describes nearby landmark without directing any movement); unlike `verify`, *stand* is only used in these cases, rather than as an anchor for landmarks
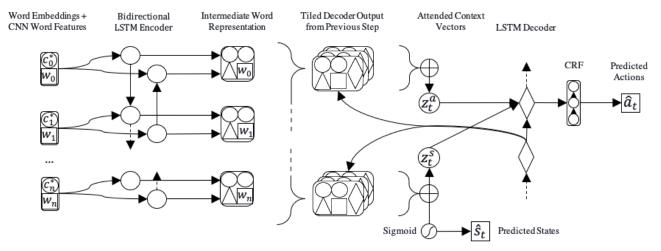
Figure 2: Full system architecture for predicting sequences of actions and states from a given sentence; word and character embeddings are passed to an encoder; outputs are concatenated with word features and combined with tiled decoder outputs; attention mechanism over word representations for each time step produces context vectors; learned attention parameters predict states; context vectors are passed to a decoder to predict actions

move actions for cases where the exact movement is unclear from the text.

Landmarks are noun phrases that describe a physical object useful in the navigation task. World state s_t represents the possibly empty set of text spans in the input sentence $\mathbf{w}$ that refer to landmarks near the tth decision point. We represent s_t as a binary vector of the same length as the input sequence ($s_t \in \{0,1\}^n$), where w_i is a token that refers to a landmark near step t *iff* $s_{ti} = 1$.

Graph Construction

Given action and state sequences $\mathbf{a}$ and $\mathbf{s}$, the landmark graph construction process is shown in Algorithm 1, where T refers to the length of the sequence.

```
Add root node: n_0
For t=1 to T:
   Add new step node: n_t
   Add labeled edge: (n_{t-1}, a_t, n_t)
   For each landmark l specified in s_t:
      If l is not in the graph:
         Add new landmark node: l
      Add edge: (l, n_t)
```

Algorithm 1: Landmark Graph Construction from Actions ($a_t \in \mathbf{a}$) and States ($s_t \in \mathbf{s}$)

As defined here, landmark graphs contain only landmark proximity information (i.e. an unlabeled edge between a landmark and decision point indicate nearness). While the rest of this paper assumes that landmark graphs will take this form, it is conceptually trivial to label these links with relative directions (e.g. in_front_of) to increase the expressivity of the graph.

Though landmark graphs are similar to the graphical representation introduced in Chen and Mooney (2011), our representation – informed by the landmark and navigation liter- ature – communicates locations with nodes and spatial relations with edges, resulting in a cleaner, more intuitive connection between the described space and the navigation task.

Approach

The central intuition behind our approach is that in order to understand a route instruction, a sentence must be parsed multiple times, focusing on different phrases on each pass to construct a spatial representation. A route is thereby decomposed into decision points; at each, our models ask *where am I* – which landmarks are near the current position – and *how did I get here* – which action was taken to traverse from the previous step to the current – all while keeping track of where it's been. To do this, the model encodes the sentence and decodes a sequence of actions, at each step attending to the parts of the sentence that inform the action and separately attending to the parts of the sentence that describe nearby landmarks.

Specifically, we apply an encoder-decoder approach to predict action sequence $\hat{\mathbf{a}}$ and state sequence $\hat{\mathbf{s}}$ from route instruction $\mathbf{w}$, where the length of the output sequences are bounded by the prediction of the *STOP* action or by reaching the maximum sequence length. Given $\hat{\mathbf{a}}$ and $\hat{\mathbf{s}}$, landmark graphs are constructed following Algorithm 1.

The full network architecture is shown in Figure 2

Encoder

We represent words with the combination of learned character-level features and dense embeddings. Specifically, each character c_i^j in word w_i is represented as a one-hot-vector combined with explicit features (e.g. is_digit, is_alpha). The word's character representations are passed to a convolutional neural network, where filters are swept over character groupings and max-pooling is applied to the filter outputs to produce fixed-length

character-level features for the word: c_i^*. These features are then concatenated with the pre-trained Word2Vec embedding (Mikolov et al. 2013) for w_i. These word representations are then passed to a bidirectional long-term-short-term-memory network (LSTM) (Hochreiter and Schmidhuber 1997) to encode the word in its context.

Decoder

Following Mei et al. (2016), we produce an action context vector z_t^a for each time step, first by combination of forward and backward encoder outputs $e_i := [e_i^{fw}, e_i^{bw}]$, the word embedding w_i, and the decoder hidden state from the previous time step d_{t-1} (tiled across input words). We also concatenate simple explicit word features ϕ_i to this representation (e.g. part-of-speech tags). This combined representation is then attended over, such that the context vector $z_t^a = \sum_i \alpha_{ti}^a [w_i, e_i, \phi_i]$, where attention weights α_{ti}^a are learned according to the following:

$$\alpha_{ti}^a = \frac{\exp(\beta_{ti}^a)}{\sum_i \exp(\beta_{ti}^a)} \qquad (1)$$
$$\beta_{ti}^a = v^a \tanh(W^a d_{t-1} + U^a w_i + V^a[e_i, \phi_i])$$

where v^a, W^a, U^a, and V^a are learned parameters. Unlike Mei et al. (2016), however, our system does not concatenate a ground truth world state to this action context vector. Instead, we learn a world state context vector z_t^s using the technique defined above. The combined vector $z_t := [z_t^a, z_t^s]$ is then passed to a unidirectional LSTM decoder.

Prediction

The decoder output is passed to a linear chain conditional random field layer (CRF) which finds the best action, $\hat{a}_t$, based both on the label scores and on the predictions for previous time steps. When predicting a world state, $\hat{s}_t$, we pass the vector $[\beta_{t0}^s, \beta_{t1}^s, ... \beta_{tn}^s]$ to a sigmoid function and consider all positive values as indicative that a word describes a landmark; these words are then grouped naively into spans.

Though learning $\hat{s}_t$ directly from β_{ti}^s is a relatively simple modification, the conceptual novelty is important to explore. In using attention to create action context vector z_t^a, the system is learning β_{ti}^a to determine whether w_i is helpful to predict action $\hat{a}_t$. We apply this same insight to the world state representation. At each time step, we assume some number of words (≥ 0) refer to landmarks near that step in the path; β_{ti}^s thus serves as a score for whether the w_i refers to a landmark for step t.

We learn separate context vectors z_t^a and z_t^s specifically because while β_{ti}^a and β_{ti}^s are learning to attend to parts of the input sentence based on the current position in the predicted path, their goals are different; words that indicate which action to take (e.g. "then turn left") are not the same words that indicate landmarks (e.g. "the corner of the house"). Both vectors are necessary for the decoder, however, as knowing the previous action and nearby landmarks is necessary to understand the current location.

Training

During training, we use negative log likelihood loss for actions: $\mathcal{L}^a = -\sum \log P(\hat{a}|\mathbf{z})$. The CRF probability for an action given a context vector is given in Equation 2, where μ_t^a is the unary score for context vector z_t taking action a_t, the probability that action a_{t+1} follows action a_t is $\theta_{a_t, a_{t+1}}$, and ζ is a normalization term (the sum of combinations over all possible actions at each time step).

$$P(\hat{a}|\mathbf{z}) = \frac{1}{\zeta} \exp\left(\sum_t^T \mu_t^a + \sum_t^{T-1} \theta_{a_t, a_{t+1}}\right) \qquad (2)$$

For states we use sigmoid cross entropy loss shown in Equation 3, where s_{ti} refers to the binary world state label for word w_i at time step t

$$\mathcal{L}^s = -\sum_t \sum_i \max(\beta_{ti}^s, 0) - \beta_{ti}^s s_{ti} +$$
$$\log(1 + \exp(-|\beta_{ti}^s|)) \qquad (3)$$

We train our model using joint loss $\mathcal{L} = \mathcal{L}^a + \mathcal{L}^s$.

Experiments

Ideally, we would evaluate our system by measuring to what extent generated landmark graphs were helpful in real-world human navigation tasks. In practice, however, we must focus on whether the generated graphs contain the same information as those of the ground truth, either through the constituent elements – actions and states – or by comparing the complete graphs.

Data

In our experiments, we train and evaluate models with two datasets. The SAIL route instruction dataset (MacMahon, Stankiewicz, and Kuipers 2006) contains three maps and natural language route instructions annotated with actions[3], states[4], and path coordinates. Since our approach operates over surface realizations rather than the fixed set of entities in SAIL (e.g. "the red brick" or "the brick alley" instead of BRICK HALLWAY), we augment their annotations with a by-sentence mapping from entities to surface strings using a heuristic approach reviewed by annotators.

While the SAIL dataset is the most appropriate publicly available dataset for our task, the instructions describe simple virtual worlds. Since we are motivated by real-world environments such as those encountered by package handlers, we collected a toy dataset of route instructions (82 paths; 188 sentences) that begin at some referential point (a street near an address) and end at a final delivery location. We refer to this as our *Delivery* dataset.

Despite being significantly smaller than SAIL (12% as many routes; 6% as many sentences), our Delivery dataset has about as large of a vocabulary (Delivery: 481; SAIL: 587) and much longer sentences (Delivery: 17.1; SAIL: 7.8). Our Delivery dataset is also more referential; where SAIL contains 527 landmark surface realizations (0.8 per

---

[3]SAIL's travel, turn left, turn right, and NULL actions can be trivially transformed to our *forward*, *left*, *right*, and *stand*; travel(step: n) is interpreted as *n forward* actions

[4]SAIL landmark entities are associated with a verify action; we attach these landmarks directly to decision points.

	MLA	Ours (SAIL)	Ours (Delivery)
Acc.	68.3%	90.6%	68.9%

Table 1: Action prediction accuracy

Distance	0	1	2	3
MLA				
Sent.	49.9%	55.8%	77.5%	85.4%
Route	14.6%	22.1%	27.7%	33.8%
Ours (SAIL)				
Sent.	88.6%	94.7%	99.3%	99.7%
Route	50.2%	56.8%	64.8%	66.7%
Ours (Delivery)				
Sent.	60.1%	88.8%	98.6%	100.0%
Route	37.0%	70.4%	85.2%	85.2%

Table 2: Goal position accuracy as a function of distance for single-sentence and full route instruction action sequences

sentence; 3.7 per route), our dataset contains 324 (2.5 per sentence; 5.6 per route). While this dataset is significantly smaller than SAIL (which is itself much smaller than corpora traditionally used in neural systems), we believe these experiments can provide important insights into our approach's applicability to real-world environments.

Experimental Setup

We use convolutional filters of size 2, 4, 8, and 16 over characters to produce character-level word feature vectors of length 16, pretrained Word2Vec embeddings of size 300, encoder hidden states of width 200, 20% dropout on encoder inputs and 50% dropout on encoder outputs. The decoder hidden state width is 256, and a batch size of 1. We train for 50 epochs using an exponentially decaying learning rate (initial: 0.001; rate: 0.99; steps: 1000) in conjunction with the Adam optimizer (Kingma and Ba 2014).

In the following sections, all reported quantitative results are three-fold averages of models trained on train + development data and evaluated on test data (where the train/dev/test split is 80/10/10). The qualitative examples shown in Figures 3 and 4 are predictions on dev. data made by models trained on the corresponding train fold. Similarly, the hyperparameters were tuned on one of the SAIL train folds and evaluated on the corresponding dev. fold.

Results

We evaluate our approach in three ways: by the predicted actions, landmark spans, and complete landmark graphs. As in previous work, we treat sentences independently from one another. Where route-level measures are shown, the sentence-level predictions were combined linearly, where all root nodes except for the first are dropped, and any *stand* action at the beginning of a sentence-level graph is replaced with *move* (as such an action indicates an uncertain anchoring at a landmark).

Using these measures, we evaluate the performance of two separate models trained on the SAIL and Delivery datasets, respectively. In order to compare our approach to similar systems, we treat the Multi-Level Aligner model (MLA) from Mei et al. (2016) as our baseline, training and evaluating their system[5] on the same folds as our SAIL model. We focus specifically on MLA – rather than more recent approaches like (Anderson et al. 2018) and (Chen et al. 2019) – because it is the approach on which ours is based. The differences in performance can thus be seen as an approximate measure of the usefulness in enabling the system to learn to find landmark spans, rather than being given a simple world state representation at inference time.

Actions

We borrow two action prediction evaluation measures from the robotic navigation literature: whether the predicted action was correct at each step (accuracy) and whether an agent following those actions arrived as a goal position given a distance threshold (distance).

While the SAIL data contains ground truth path coordinates, our Delivery data – and any similarly constructed real-world dataset – does not. We therefore must handle ambiguous actions: *move → forward*, and *turn* has a 50/50 chance of being *right* or *left*. This randomness should have minimal effects given our three-fold validation and the rarity of *turn* actions, but this means the distance measure on the Delivery data is less consistent.

It's important to note that our measures are distinct from previous work in two ways. First, we entirely disregard orientation, as our simplified landmark graph representation only considers a landmark's nearness, rather than its relative direction. Second, we capture cases where a final position is close to but not matching a goal position with Euclidean distance, rather than the number of intersections in the SAIL grid. While the overall effect of this discrepancy should be minimal, it makes it difficult to compare our work directly with previous papers like (Artzi and Zettlemoyer 2013) and (Andreas and Klein 2015).

The action accuracy and distance results are shown in Tables 1 and 2, respectively.

Our SAIL model significantly outperforms the MLA baseline (+23.3% action accuracy; +38.7% single sentence goal accuracy; +35.6% route goal accuracy) on which it is based, suggesting that jointly learning to identify nearby landmarks helps the model predict actions.

The performance of our Delivery model is more modest, suggesting both the increased difficulty of that setting and of training a model on so few examples. Our Delivery model does exhibit the same accuracy increase as a function of distance as both our SAIL model and the MLA baseline, suggesting that the model is still learning to capture spatial relationships in this setting (particularly when permitting a distance threshold of 1).

---

[5]Our MLA baseline results are actions and states produced by the publicly available code trained for 50 epochs.

	J	P	R	F1
Ours (SAIL)				
Step	71.6%	0.0%	0.0%	0.0%
Sent.	68.9%	30.7%	31.8%	31.0%
Route	63.7%	44.8%	46.2%	44.6%
Ours (Delivery)				
Step	39.6%	0.0%	0.0%	0.0%
Sent.	40.3%	6.5%	6.7%	6.3%
Route	39.6%	8.3%	10.2%	9.0%

Table 3: Landmark span prediction performance measured across steps (correct spans at the right step), sentences (correct spans for a sentence), and routes (correct spans for the route instruction)

	Sent. $(\text{sim}|\text{sim}^\ell)$		Route $(\text{sim}|\text{sim}^\ell)$	
MLA	76.6%		65.9%	
Ours (SAIL)	91.1%	92.9%	83.1%	88.0%
Ours (Delivery)	67.6%	73.3%	57.0%	63.3%

Table 4: Graph similarity for sentences and routes using the strict (sim) and relaxed (sim^ℓ) measures

Landmarks

We evaluate the performance of our models on the landmark span detection task in two ways. In the more traditional measure, we compare the predicted landmark spans with the ground truth using precision, recall, and F1, where a predicted span is correct *iff* it matches the ground truth exactly. Our more permissive measure compares the tokens of the predicted and ground truth spans using the Jaccard index: $J(l^p, l^g) = |l^p \cap l^g|/|l^p \cup l^g|$, where l refers to the set of tokens in a span ($w \in l$).

We evaluate the ability of our models to identify the landmark spans across steps, sentences, and routes. These results are shown in Table 3. Since the MLA baseline does not predict landmark spans, no results are shown.

For both sentences and routes, our SAIL model finds approximately the correct range of tokens that refer to landmarks (evidenced by the high Jaccard index) and the exact span (shown by the F1 score). While these scores are poor in comparison to modern methods for mention detection[6], they show that a fully end-to-end approach for capturing both spatial relations and landmark information in one system is beginning to yield positive results.

Though our Delivery model behaves similarly to our SAIL model, the performance is significantly worse (likely due to the small dataset). It's also worth noting that for both models the step-level F1 is approximately 0 despite a similar Jaccard index for steps and sentences. This is likely because when a landmark span is predicted it overlaps meaningfully with the ground truth, but the vast majority of steps have no
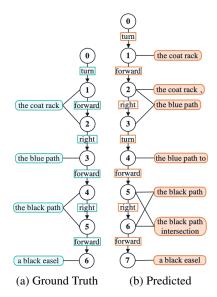
---

[6] Peng et al. (2015) reports mention detection scores in the 70-90% range

(a) Ground Truth   (b) Predicted

Figure 3: Ground truth (aqua) and predicted (orange) landmark graphs for SAIL route: "Go towards the coat rack. At the coat rack, take a right onto the blue path. Follow the blue path to the black path intersection and go right onto the black path. Go all the way down until you get to a black easel."

predicted span and thus the resulting average is near zero.

Landmark Graphs

Though the decomposition of spatial representation prediction into states and actions is useful both for the model and for indirect evaluation purposes, our goal is to extract complete spatial representations from route instructions. Therefore, we compare predicted landmark graph g^p against the ground truth graph g^g using a modification of graph edit distance (Abu-Aisheh et al. 2015). This is defined in Equation 4

$$\text{sim}(g^p, g^g) = 1 - \frac{1}{|g^p| + |g^g|} \left(\min_{\gamma \in \Gamma(g^p, g^g)} \sum_i c(\gamma_i) \right) \quad (4)$$

where $|g|$ refers to the sum of the number of edges and number of nodes in graph g, $\Gamma(g^p, g^g)$ refers to the set of possible edit paths transforming g^p to g^g, γ_i is an edit operation in path γ, and c is the cost of that operation.

In our strict measure, sim, the insertion and deletion operations have a cost of 1. Substitution costs 0 if the attributes of the two nodes are the same (i.e. an edge labeled with an action can freely replace an edge with the same label, and a landmark node can be substituted for a landmark node with the exact same string), and otherwise costs 1. This measure thus corresponds similarity: the percentage of possible edits that were not necessary in transforming g^p to g^g.

Our more permissive measure, $\text{sim}^\ell(g^p, g^g)$, sets the landmark node substitution cost at $1 - J(l^p, l^g)$, allowing graphs to be penalized less for inexact landmark matches.

In order to make the comparison to the MLA baseline as

(a) Ground Truth   (b) Predicted
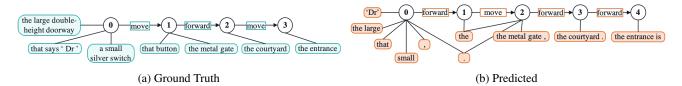
Figure 4: Ground truth (aqua) and predicted (orange) landmark graphs for Delivery route: "To gain access look for the large double-height doorway, and on the right hand side there's a small silver switch that says 'Dr'. Press that button and then push the metal gate, to walk through into the courtyard. The entrance is on the right hand side."

fair as possible, we use a by-sentence mapping[7] to replace ground truth entities with surface realizations. Note, however, that the MLA approach is aware of all ground truth entities for each step, not just referents for spans in the instruction; this means their graphs tend to overgenerate for our setting, leading to poorer performance.

Our graph similarity results are shown in Table 4, where MLA's sim=sim^ℓ, since inexact matches aren't possible.

As in our other measures, our SAIL model outperforms the MLA baseline while the performance of the Delivery model is significantly lower. Most importantly, though, this measure shows that the spatial representations predicted by these models have a high degree of similarity with the ground truth graphs, particularly when span approximation, rather than exact matching, is incorporated into the measure.

Examples

SAIL An example predicted landmark graph from our SAIL model is shown in Figure 3. These results are nearly perfect, though there's an additional *turn* after the *right* (along with a corresponding decision point), and the predicted landmark spans are (at worst) minor variations on the ground truth.

One important limitation is the absence of landmark coreference resolution: since landmarks must be found anew at each step, the prediction of "the coat rack ," at step 2 is interpreted as a distinct span from "the coat rack". Another interesting phenomena is the prediction of "the black path intersection" and its association with the same decision points as "the black path". While this is strictly incorrect, a valid interpretation of the route instruction may include this landmark, meaning that the model discovered a useful span that was missed during annotation.

Delivery Figure 4 shows a predicted from graph from our Delivery model. Here, the model attended to the parts of the sentence referring to landmarks (including the gate and the courtyard) while also capturing the appropriate spatial relations as expressed by actions (neither the predicted nor the ground truth graph have any turns).

However, this example demonstrates limitations in both our task framing and our models' capabilities. Conceptually, spatial relations are semantically fuzzy (e.g. two *forward* actions may refer to different real-world distances). In

the context of these graphs, this means that while the represented spatial relationships are very similar (and would be interpreted as such by a person) it is still difficult to measure this similarity automatically or for a robot to interpret these actions. Where landmarks are concerned, it's clear that while spans are found in roughly the right locations at the right steps, finding exact span boundaries is difficult for the model: "the large" and "small" are missing the most important tokens in their spans, while "the metal gate ," and "the entrance is" contain spurious tokens.

Overall, these predicted graphs confirm that our approach is producing a fairly accurate representations of spaces described by route instructions, but more work is needed.

Conclusion

We have introduced the task of landmark graph generation and an approach to create these spatial representations by jointly predicting landmark spans and traversal actions. We show that our models yield good performance according to the graph similarity measure we introduce, as well as the related action prediction evaluation measures borrowed from the robotic navigation literature.

However, it is also clear from our results that this work is a first step in need of refinement. Landmark span detection in particular suffers from the simplicity of our approach, and future work will likely incorporate insights from the grounding and coreference resolution literature (particularly approaches like Lee et al. (2017)).

We believe that the landmark graph generation task to be a critical next step in the development of landmark-centric navigation technologies, and the results of our Delivery model point to the complexity of the real-world domain and the need for large datasets that capture this complexity.

References

[2015] Abu-Aisheh, Z.; Raveaux, R.; Ramel, J.-Y.; and Martineau, P. 2015. An exact graph edit distance algorithm for solving pattern recognition problems.

[2018] Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and van den Hengel, A. 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*, 3674–3683.

[2015] Andreas, J., and Klein, D. 2015. Alignment-based

---

[7]The entities in the MLA code differ slightly from those in SAIL proper; we therefore constructed a new mapping in the same way we did for SAIL.

compositional semantics for instruction following. *arXiv preprint arXiv:1508.06491*.

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

[2011] Chen, D. L., and Mooney, R. J. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Conference for the Association for the Advancement of Artificial Intelligence (AAAI)*, volume 2.

[2019] Chen, H.; Suhr, A.; Misra, D.; Snavely, N.; and Artzi, Y. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12538–12547.

[2016] Duvallet, F.; Walter, M. R.; Howard, T.; Hemachandra, S.; Oh, J.; Teller, S.; Roy, N.; and Stentz, A. 2016. Inferring maps and behaviors from natural language instructions. In *Experimental Robotics*.

[2017] Fellner, I.; Huang, H.; and Gartner, G. 2017. turn left after the wc, and use the lift to go to the 2nd floorgeneration of landmark-based route instructions for indoor navigation. *ISPRS International Journal of Geo-Information* 6(6):183.

[1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

[2016] Kai-Florian, R., and Winter, S. 2016. *Landmarks: Giscience for Intelligent Services*. Springer.

[2015] Kennington, C., and Schlangen, D. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

[2014] Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[2010] Kollar, T.; Tellex, S.; Roy, D.; and Roy, N. 2010. Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, 259–266. IEEE Press.

[2014] Kong, C.; Lin, D.; Bansal, M.; Urtasun, R.; and Fidler, S. 2014. What are you talking about? text-to-image coreference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3558–3565.

[2013] Krishnamurthy, J., and Kollar, T. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics (TACL)* 1:193–206.

[2017] Landsiedel, C.; Rieser, V.; Walter, M.; and Wollherr, D. 2017. A review of spatial reasoning and interaction for real-world robotics. *Advanced Robotics* 31(5):222–242.

[2017] Lee, K.; He, L.; Lewis, M.; and Zettlemoyer, L. 2017. End-to-end neural coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 188–197.

[2006] MacMahon, M.; Stankiewicz, B.; and Kuipers, B. 2006. Walk the talk: Connecting language, knowledge, and action in route instructions. *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence* 2(6):4.

[2016] Mei, H.; Bansal, M.; and Walter, M. R. 2016. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Proceedings of the Conference for the Association for the Advancement of Artificial Intelligence (AAAI)*, volume 1.

[2013] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

[2015] Peng, H.; Chang, K.-W.; and Roth, D. 2015. A joint framework for coreference resolution and mention head detection. 51:12.

[2015] Plummer, B. A.; Wang, L.; Cervantes, C.; Caicedo, J. C.; Hockenmaier, J.; and Lazebnik, S. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2641–2649.

[2017] Plummer, B. A.; Mallya, A.; Cervantes, C.; Hockenmaier, J.; and Lazebnik, S. 2017. Phrase localization and visual relationship detection with comprehensive linguistic cues. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[2006] Tsetsos, V.; Anagnostopoulos, C.; Kikiras, P.; and Hadjiefthymiades, S. 2006. Semantically enriched navigation for indoor environments. *International Journal of Web and Grid Services* 2(4):453–478.

[2015] Yang, L., and Worboys, M. 2015. Generation of navigation graphs for indoor space. *International Journal of Geographical Information Science* 29(10):1737–1756.

[2018] Zang, X.; Vázquez, M.; Niebles, J. C.; Soto, A.; and Savarese, S. 2018. Behavioral indoor navigation with natural language directions. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 283–284. ACM.