Learning to Read Maps: Understanding Natural Language Instructions from Unseen Maps

Miltiadis Marios Katsakioris[1], Ioannis Konstas[1], Pierre Yves Mignotte[2], Helen Hastie[1]
[1]School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, UK
[2]SeeByte Ltd, Edinburgh, UK
Mmk11, I.konstas, H.hastie@hw.ac.uk
Pierre-yves.mignotte@seebyte.com

Abstract

Robust situated dialog requires the ability to process instructions based on spatial information, which may or may not be available. We propose a model, based on LXMERT, that can extract spatial information from text instructions and attend to landmarks on OpenStreetMap (OSM) referred to in a natural language instruction. Whilst, OSM is a valuable resource, as with any open-sourced data, there is noise and variation in the names referred to on the map, as well as, variation in natural language instructions, hence the need for data-driven methods over rule-based systems. This paper demonstrates that the gold GPS location can be accurately predicted from the natural language instruction and metadata with 72% accuracy for previously seen maps and 64% for unseen maps.

USER: Send one drone 89m south west of Chevron to put out the fire.

Landmark: Chevron
Distance: 89
Bearing: S-W

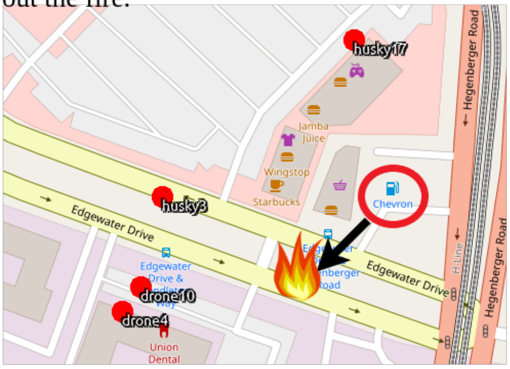


Figure 1: User instruction and the corresponding image, displaying 4 robots and landmarks. The users were not restricted or prompted to use specific landmarks on the map. The circle around the target landmark was added for clarity for this paper; users were not given any such visual hints.

1 Introduction

Spoken dialog systems are moving into real world situated dialog, such as assisting with emergency response and remote robot instruction that require knowledge of maps or building schemas. Effective communication of such an intelligent agent about events happening with respect to a map requires learning to associate natural language with the world representation found within the map. This symbol grounding problem (Harnad, 1990) has been largely studied in the context of mapping language to objects in a situated simple (MacMahon et al., 2006; Johnson et al., 2017) or 3D photorealistic environments (Kolve et al., 2017; Savva et al., 2019), static images (Ilinykh et al., 2019; Kazemzadeh et al., 2014), and to a lesser extent on synthetic (Thompson et al., 1993) and real geographic maps (Paz-Argaman and Tsarfaty, 2019; Haas and Riezler, 2016; Götze and Boye, 2016). The tasks usually relate to navigation (Misra et al., 2018; Thomason et al., 2019) or action execution (Bisk et al., 2018; Shridhar et al., 2019) and assume giving instructions to an embodied egocentric agent with a shared first-person view. Since most rely on the visual modality to ground natural language (NL), referring to items in the immediate surroundings, they are often less geared towards the accuracy of the final goal destination.

The task we address here is the prediction of the GPS of this goal destination by reference to a map, which is of critical importance in applications such as emergency response where specialized personnel or robots need to operate on an exact location (see Fig. 1 for an example). Specifically, the goal we are trying to predict is in terms of: a) the GPS coordinates (latitude/longitude) of a referenced landmark; b) a compass direction (bearing) from this referenced landmark; and c) the distance in meters from the referenced landmark. This is done by taking as input into a model: i) the knowledge base of the symbolic representation of the world such as landmark names and regions of interest (metadata); ii) the graphic depiction of a map

Proceedings of Second International Combined Workshop on Spatial Language Understanding andGrounded Communication for Robotics, pages 11–21
August 5–6, 2020. ©2021 Association for Computational Linguistics

(visual modality); and iii) a worded instruction.

Our approach to the destination prediction task is two-fold. The first stage is a data collection for the "Robot Open Street Map Instructions" (ROSMI) (Katsakioris et al., 2020) corpus based on OpenStreetMap (Haklay and Weber, 2008), in which we gather and align NL instructions to their corresponding target destination. We collected 560 NL instruction pairs on 7 maps of different variety and landmarks, in the domain of emergency response using Amazon Mechanical Turk. The subjects are given a scene in the form of a map and are tasked to write an instruction to command a conversational assistant to direct robots and autonomous systems to either inspect an area or extinguish a fire. The setup was intentionally emulating a typical *'Command and Control'* interface found in emergency response hubs, in order to promote instructions that accurately describe the final destination, with regards to its surrounding map entities.

Whilst OSM and other crowdsourced resources are hugely valuable, there is an element of noise associated with the metadata collected in terms of the names of the objects on the map, which can vary for the same type of object (e.g. newsagent/kiosk, confectionary/chocolate store etc.), whereas the symbols on the map are from a standard set, which one hypothesizes a vision-based trained model could pick-up on. To this end, we developed a model that leverages both vision and metadata to process the NL instructions.

Specifically, our MAPERT (Map Encoder Representations from Transformers) is a Transformer-based model based on LXMERT. It comprises of up to three single-modality encoders for each input (i.e., vision, metadata and language), an early fusion of modalities components and a cross-modality encoder, which fuses the map representation (metadata and/or vision) with the word embeddings of the instruction in both directions, in order to predict the three outputs, i.e., reference landmark location on the map, bearing and distance.

Our contributions are thus three-fold:

- A novel task for final GPS destination prediction from NL instructions with accompanying ROSMI dataset[1].
- A model that predicts GPS goal locations from a map-based natural language instruction.
- A model that is able to understand instructions referring to previously unseen maps.

2 Related Work

Situated dialog encompasses various aspects of interaction. These include: situated Natural Language Processing (Bastianelli et al., 2016); situated reference resolution (Misu, 2018); language grounding (Johnson et al., 2017); visual question answer/visual dialog (Antol et al., 2015); dialog agents for learning visually grounded word meanings and learning from demonstration (Yu et al., 2017); and Natural Language Generation (NLG), e.g. of situated instructions and referring expressions (Byron et al., 2009; Kelleher and Kruijff, 2006). Here, work on instruction processing for destination mapping and navigation are discussed, as well as language grounding and referring expression resolution, with an emphasis on 2D/3D real world and map-based application.

Language grounding refers to interpreting language in a situated context and includes collaborative language grounding toward situated human-robot dialog (Chai et al., 2016), city exploration (Boye et al., 2014), as well as following high-level navigation instructions (Blukis et al., 2018). Mapping instructions to low level actions has been explored in structured environments by mapping raw visual representations of the world and text onto actions using using Reinforcement Learning methods (Misra et al., 2017; Xiong et al., 2018; Huang et al., 2019). This work has recently been extended to controlling autonomous systems and robots through human language instruction in a 3D simulated environment (Ma et al., 2019; Misra et al., 2018; Blukis et al., 2019) and Mixed Reality (Huang et al., 2019) and using imitation learning (Blukis et al., 2018). These systems perform goal prediction and action generation to control a single Unmanned Aerial Vehicles (UAVs), given a natural language instruction, a world representation and/or robot observations. However, where this prior work uses raw pixels to generate a persistent semantic map from the system's line-of-sight image, our model is able to leverage both pixel and metadata, when it is available in a combined approach. Other approaches include neural mapping of navigational instructions to action sequences (Mei et al., 2015), which does include a representation of the observable world state, but this is more akin to a maze rather than a complex map.

[1] We make our code and data available at https://github.com/marioskatsak/mapert.

With respect to the task, our model looks to predict GPS locations. There are few related works that attempt this challenging task. One study, as part of the ECML/PKDD challenge (de Brébisson et al., 2015), uses Neural Networks for Taxi Destination Prediction as a sequence of GPS points. However, this does not include processing natural language instructions. SPACEREF (Götze and Boye, 2016) is perhaps the closest to our task in that the task entails both GPS tracks in OSM and annotated mentions of spatial entities in natural language. However, it is different in that these spatial entities are viewed and referred to in a first person view, rather than entities on a map (e.g. "the arch at the bottom").

In terms of our choice of model, attention mechanisms (Bahdanau et al., 2015; Vaswani et al., 2017; Xu et al., 2015) have proven to be very powerful in language and vision tasks and we draw inspiration from the way (Xu et al., 2015) use attention to solve image captioning by associating words to spatial regions within a given image.

3 Data

As mentioned above, the task is based on OpenStreetMap (OSM) (Haklay and Weber, 2008). OSM is a massively collaborative project, started in 2004, with the main goal to create a free editable map of the world. The data is available under the Open Data Commons Open Database Licence and has been used in some prior work (Götze and Boye, 2016; Hentschel and Wagner, 2010; Haklay and Weber, 2008). It is a collection of publicly available geodata that are constantly updated by the public and consists of many layers of various geographic attributes of the world. Physical features such as roads or buildings are represented using tags (metadata) that are attached to its basic data structures. A comprehensive list of all the possible features available as metadata can be found online[2]. There are two types of objects, *nodes* and *ways*, with unique IDs that are described by their latitude/longitude (lat/lon) coordinates. Nodes are single points (e.g. coffee shops) whereas ways can be more complex structures, such as polygons or lines (e.g. streets and rivers). For this study, we train and test only on data that uses single points (nodes) and polygons (using the centre point), and leave understanding more complex structures as future work.

We train and evaluate our model on ROSMI, a new multimodal corpus. This corpus consists of visual and natural language instruction pairs, in the domain of emergency response. In this data collection, the subjects were given a scene in the form of an OSM map and were tasked to write an instruction to command a conversational assistant to direct a number of robots and autonomous systems to either inspect an area or extinguish a fire. Figure 1 shows an example of such a written instruction. These types of emergency scenarios usually have a central hub for operators to observe and command humans and Robots and Autonomous Systems (RAS) to perform specific functions, where the robotic assets are visually observable as an overlay on top of the map. Each instruction datapoint was manually checked and if it did not match the 'gold standard' GPS coordinate per the scenario map, it was discarded. The corpus was manually annotated with the ground truth for, (1) a link between the NL instruction and the referenced OSM entities; and (2) the distance and bearing from this referenced entity to the goal destination. The ROSMI corpus thus comprises 560 tuples of instructions, maps with metadata and target GPS location.

There are three linguistic phenomena of note that we observe in the data collected. Firstly, **Landmark Grounding** where each scenario has 3-5 generated *robots* and an average of 30 *landmarks* taken from OSM. Each subject could refer to any of these objects on the map, in order to complete the task. Grounding the right noun phrase to the right OSM landmark or robot, is crucial for predicting accurately the gold-standard coordinate, e.g. *send husky11 62m to the west direction* or *send 2 drones near Harborside Park*.

Secondly, **Bearing/Distance** factors need to be extracted from the instruction such as numbers (e.g. 500 meters) and directions (e.g. northwest, NE) and these two items typically come together. For example, *"send drone11 to the west about 88m"*.

Thirdly, **Spatial Relations** are where prepositions are used instead of distance/bearing (e.g. near, between), and are thus more vague. For example, *"Send a drone near the Silver Strand Preserve"*.

4 Approach

4.1 Task Formulation

An instruction is taken as a sequence of word tokens $\mathbf{w} =< w_1, w_2, \dots w_N >$ with $\mathbf{w_i} \in V$,

[2] wiki.openstreetmap.org/wiki/Map_Features

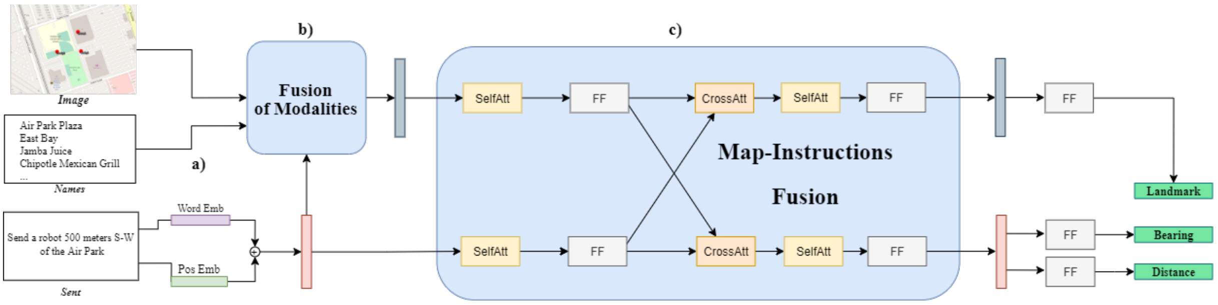


Figure 2: Architecture of MAPERT. Map representations, i.e., names of landmarks found in OSM (metadata) and Faster-RCNN predicted objects (visual modality), along with an instruction (sequence of tokens) are a) encoded into the model, b) fused together (see also Fig. 4) and c) bidirectionally attended. The output comprises of three predictions, recast as classification tasks: a landmark, a bearing and a distance.

where V is a vocabulary of words and the corresponding geographic map I is represented as a set of M landmark objects $o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{n})$ where $\mathbf{bb}$ is a 4-dimensional vector with bounding box coordinates, $\mathbf{r}$ is the corresponding Region of Interest (RoI) feature vector produced by an object detector and $n =< n_1, n_2 \dots n_K >$, is a multi-token name. We define a function $f : V^N \times R^{4*M} \times R^{2048*M} \times V^{M*K} \rightarrow R \times R$ to predict the GPS destination location $\hat{y}$:

$$\hat{y} = f\big(\mathbf{w}, \{o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{n})\}_M\big) \qquad (1)$$

Since predicting $\hat{y}$ directly from $\mathbf{w}$ is a harder task, we decompose it into three simpler components, namely predicting a reference *landmark* location $l \in M$, the compass direction (bearing) b[3], and a distance d from l in meters. Then we trivially convert to the final GPS position coordinates. Equation 1 now becomes:

$$\hat{y} = gps(l, d, b) = f\big(\mathbf{w}, \{o_i = (\mathbf{bb}, \mathbf{r}, \mathbf{n})\}_M\big) \qquad (2)$$

4.2 Model Architecture

Inspired by LXMERT (Tan and Bansal, 2019), we present MAPERT, a Transformer-based (Vaswani et al., 2017) model with three separate single-modality encoders (for NL instructions, metadata and visual features) and a cross-modality encoder that merges them. Fig. 2 depicts the architecture. In the following sections, we describe each component separately.

Instructions Encoder The word sequence $\mathbf{w}$ is fed to a Transformer encoder and output hidden states $\mathbf{h_w}$ and position embeddings $\mathbf{pos_w}$; its weights are initialized using pretrained BERT (Devlin et al., 2019). $\mathbf{h_{w_0}}$ is the hidden state for the special token [CLS].

Metadata Encoder OSM comes with useful metadata in the form of bounding boxes (around the landmark symbols) and names of landmarks on the map. We represent each bounding box as a 4-dimensional vector $\mathbf{bb_{meta_k}}$ and each name ($\mathbf{n_k}$) using another Transformer initialized with pretrained BERT weights. We treat metadata as a bag of names but since each word can have multiple tokens, we output position embeddings $\mathbf{pos_{n_k}}$ for each name separately; $\mathbf{h_{n_k}}$ are the resulting hidden states with $\mathbf{h_{n_{k,0}}}$ being the hidden state for [CLS].

Visual Encoder Each map image is fed into a pretrained Faster R-CNN detector (Ren et al., 2015), which outputs bounding boxes and RoI feature vectors $\mathbf{bb_k}$ and $\mathbf{r_k}$ for k objects. In order to learn better representation for landmarks, we fine-tuned the detector on around 27k images of maps to recognize k objects $\{o_1, .., o_k\}$ and classify landmarks of 213 manually-cleaned classes from OSM; we fixed k to 73 landmarks. Finally, a combined position-aware embedding $\mathbf{v_k}$ was learned by adding together the vectors $\mathbf{bb_k}$ and $\mathbf{r_k}$ as in LXMERT:

$$\mathbf{v_k} = \frac{FF(\mathbf{bb_k}) + FF(\mathbf{r_k})}{2} \qquad (3)$$

where FF are feed-forward layers with no bias.

4.3 Variants for Fusion of Input Modalities

We describe three different approaches to combining knowledge from maps with the NL instructions:

Metadata and Language The outputs of the metadata and language encoders are fused by conditioning each landmark name n_i on the instruction

[3] $b \in \{N, NE, NW, E, SE, S, SW, SE, W, None\}$.

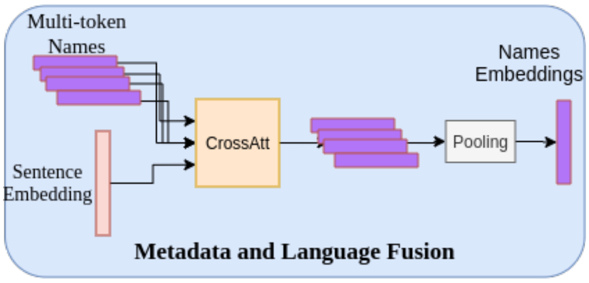


Figure 3: Metadata and Language fusion module. Multi-token names correspond to the BERT-based embeddings of landmarks names. The output is the embedding used to represent the landmarks names from OSM metadata.

sequence via a uni-directional cross attention layer (Fig. 3). We first compute the attention weights A_k between the name tokens $\mathbf{n_{k,i}}$ of each landmark o_k and instruction words in $\mathbf{h_w}$[4] and re-weight the hidden states $\mathbf{h_{n_k}}$ to get the context vectors $\mathbf{c_{n_k}}$. We then pool them using the context vector for the [CLS] token of each name:

$$\begin{aligned} \mathbf{A_k} &= CrossAttn(\mathbf{h_w}, \mathbf{n_k}) && (4) \\ \mathbf{c_{n_k}} &= \mathbf{A_k} \odot \mathbf{n_k} && (5) \\ \mathbf{h_{meta}} &= BertPooler(\mathbf{c_{n_k}}) && (6) \end{aligned}$$

We can also concatenate the bounding box $\mathbf{bb_{meta_k}}$ to the final hidden states:

$$\mathbf{h_{meta+bb}} = [\mathbf{h_{meta}}; FF(\mathbf{bb_{meta_k}})] \qquad (7)$$

Metadata+Vision and Language All three modalities were fused to verify whether vision can aid metadata information for the final GPS destination prediction task (Fig. 4). First, we filter the landmarks o_i based on the Intersection over Union between the bounding boxes found in metadata ($\mathbf{bb_{meta_k}}$) and those predicted with Faster R-CNN ($\mathbf{bb_k}$), thus keeping their corresponding names n_i and visual features $\mathbf{v_i}$. Then, we compute the instruction-conditioned metadata hidden states $\mathbf{h_{meta_i}}$, as described above, and multiply them with every object v_i to get the final $\mathbf{h_{meta+vis}}$ context vectors:

$$\mathbf{h_{meta+vis_i}} = \mathbf{h_{meta_i}} \otimes \mathbf{v_i} \qquad (8)$$

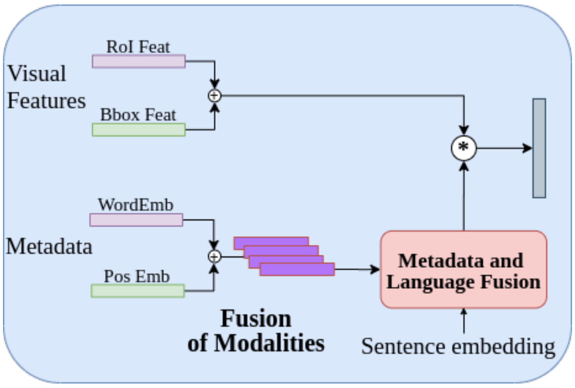


Figure 4: Fusion of metadata, vision and language modalities. Metadata are first conditioned on the instruction tokens as shown in Fig. 3. Then, they are multiplied with the visual features of every landmark.

4.4 Map-Instructions Fusion

So far we have conditioned modalities in one direction, i.e., from the instruction to metadata and visual features. In order to capture the influence between map and instructions in both ways, a cross-modality encoder was implemented (right half of Fig. 2). Firstly each modality passes through a self-attention and feed-forward layer to highlight inter-dependencies. Then these modulated inputs are passed to the actual fusion component, which consists of one bi-directional cross-attention layer, two self-attention layers, and two feed-forward layers. The cross-attention layer is a combination of two unidirectional cross-attention layers, one from instruction tokens ($\mathbf{h_w}$) to map representations (either of $\mathbf{h_{meta_k}}$, $\mathbf{v_k}$ or $\mathbf{h_{meta+vis_k}}$; we refer to them below as $\mathbf{h_{map_k}}$) and vice-versa:

$$\begin{aligned} \tilde{\mathbf{h}}_{\mathbf{w}} &= FF(SelfAtt(\mathbf{h_w})) && (9) \\ \tilde{\mathbf{h}}_{\mathbf{map_k}} &= FF(SelfAtt(\mathbf{h_{map_k}})) && (10) \\ \mathbf{C_{map_k}} &= CrossAtt(\tilde{\mathbf{h}}_{\mathbf{w}}, \tilde{\mathbf{h}}_{\mathbf{map_k}}) && (11) \\ \mathbf{C_w} &= CrossAtt(\tilde{\mathbf{h}}_{\mathbf{map_k}}, \tilde{\mathbf{h}}_{\mathbf{w}}) && (12) \\ \mathbf{h_{cross,w}} &= \mathbf{C_w} \odot \tilde{\mathbf{h}}_{\mathbf{w}} && (13) \\ \mathbf{h_{cross,map_k}} &= \mathbf{C_{map_k}} \odot \tilde{\mathbf{h}}_{\mathbf{map_k}} && (14) \\ \mathbf{out_w} &= FF(SelfAtt(\mathbf{h_{cross,w}})) && (15) \\ \mathbf{out_{map_k}} &= FF(SelfAtt(\mathbf{h_{cross,map_k}})) && (16) \end{aligned}$$

Note that representing $\mathbf{h_{map_k}}$ with vision features $\mathbf{v_k}$ only is essentially a fusion between the vision and language modalities. This is a useful variant of our model to measure whether the visual representation of a map alone is as powerful as

[4] Whenever we refer to hidden states $\mathbf{h_w}$ we assume concatenation with corresponding positional embeddings $[\mathbf{h_w}; \mathbf{pos_w}]$, which we omit here for brevity.

metadata, specifically for accurately predicting the GPS location of the target destination.

4.5 Output Representations and Training

As shown in the right-most part of Fig. 2, our MAPERT model has three outputs: landmarks, distances, and bearings. We treat each output as a classification sub-task, i.e., predicting one or the k landmarks in the map; identifying in the NL instruction the start and end position of the sequence of tokens that denotes a distance from the reference landmark (e.g., *'500m'*); and a bearing label. MAPERT's output comprises of two feature vectors, one for the vision and one for the language modality generated by the cross-modality encoder.

More specifically, for the bearing predictor, we pass the hidden state $\mathbf{out_{w,0}}$, corresponding to [CLS], to a FF followed by a softmax layer. Predicting distance is similar to span prediction for Question Answering tasks; we project each of the tokens in $\mathbf{out_w}$ down to 2 dimensions corresponding to the distance span boundaries in the instruction sentence. If there is no distance in the sentence e.g., *"Send a drone at Jamba Juice"*, the model learns to predict, both as start and end position, the final end of sentence symbol, as an indication of absence of distance. Finally, for landmark prediction we project each of the k map hidden states $\mathbf{out_{map_k}}$ to a single dimension corresponding to the index of the i^{th} landmark.

We optimize MAPERT by summing the cross-entropy losses for each of the classification sub-tasks. The final training objective becomes:

$$\mathcal{L} = \mathcal{L}_{land} + \mathcal{L}_{bear} + \mathcal{L}_{dist,start} + \mathcal{L}_{dist,end} \quad (17)$$

5 Experimental Setup

Implementation Details We evaluate our model on the ROSMI dataset and assess the contribution of the metadata and vision components as described above. For the attention modules, we use a hidden layer with size of 768 as in $BERT_{BASE}$ and we set the numbers of all the encoder and fusion layers to 1. We initialize pretrained BERT embedding layers (we also show results with randomly initialized embeddings). We trained our model using Adam (Kingma and Ba, 2015) as the optimizer with a linear-decayed learning-rate schedule (Tan and Bansal, 2019) for 90 epochs, a dropout probability of 0.1 and learning rate of 10^{-3}.

	10-fold Cross Validation (unseen examples)	
	Acc$_{50}[SD]$	T Err(m) [SD]
Oracle$_{lower}$	80 [5.01]	23.8 [51.9]
Vision		
bbox	46.18 [5.59]	44.7 [51.7]
RoI+bbox	60.36 [5.3]	36.4 [51.1]
Meta+Vision		
RoI+bbox+names	69.27 [6.68]	26.9 [47.7]
Meta		
bbox	46.18 [5.59]	44.7 [51.7]
names	**71.81 [7.37]**	26.7 [47.7]
bbox+names	70.73 [6.58]	26.3 [48.7]
Oracle$_{upper}$	100 [0.0]	0 [0]
Meta		
bbox	60.36 [5.26]	29.8 [44.9]
names	**87.64 [4.8]**	9.6 [29.9]
bbox+names	87.09 [5.66]	9.5 [27.2]

Table 1: Ablation results on ROSMI using a 10-fold cross validation. Accuracy (Acc) with IoU of 0.5 and Targer error (T Err) in meters. The results in the top half of the table use names conditioned on the lower bound of the Vision modality and so are compared to Oracle$_{lower}$. The bottom part of the table use the true metadata names and so are to be compared to Oracle$_{upper}$.

Evaluation Metrics We use a 10-fold cross-validation for our evaluation methodology. This results in a less biased estimate of the accuracy over splitting the data into train/test due to the modest size of the dataset. In addition, we performed a leave-one-map-out cross-validation, as in Chen and Mooney (2011). In other words, we use 7-fold cross-validation, and in each fold we use six maps for training and one map for validation. We refer to these scenarios as zero-shot[5] since, in each fold, we validate our data on an unseen map scenario. With the three outputs of our model, landmark, distance and bearing, we indirectly predict the destination location. Success is measured by the Intersection over Union (IoU) between the ground truth destination location and the calculated destination location. IoU measures the overlap between two bounding boxes and as in Everingham et al. (2010), must exceed 0.5 (50%) to count it as successful by the formula:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (18)$$

Since we are dealing with GPS coordinates but also image pixels, we report two error evaluation

[5] We loosely use the term zero-shot as we appreciate that there might be some overlap in terms of street names and some objects

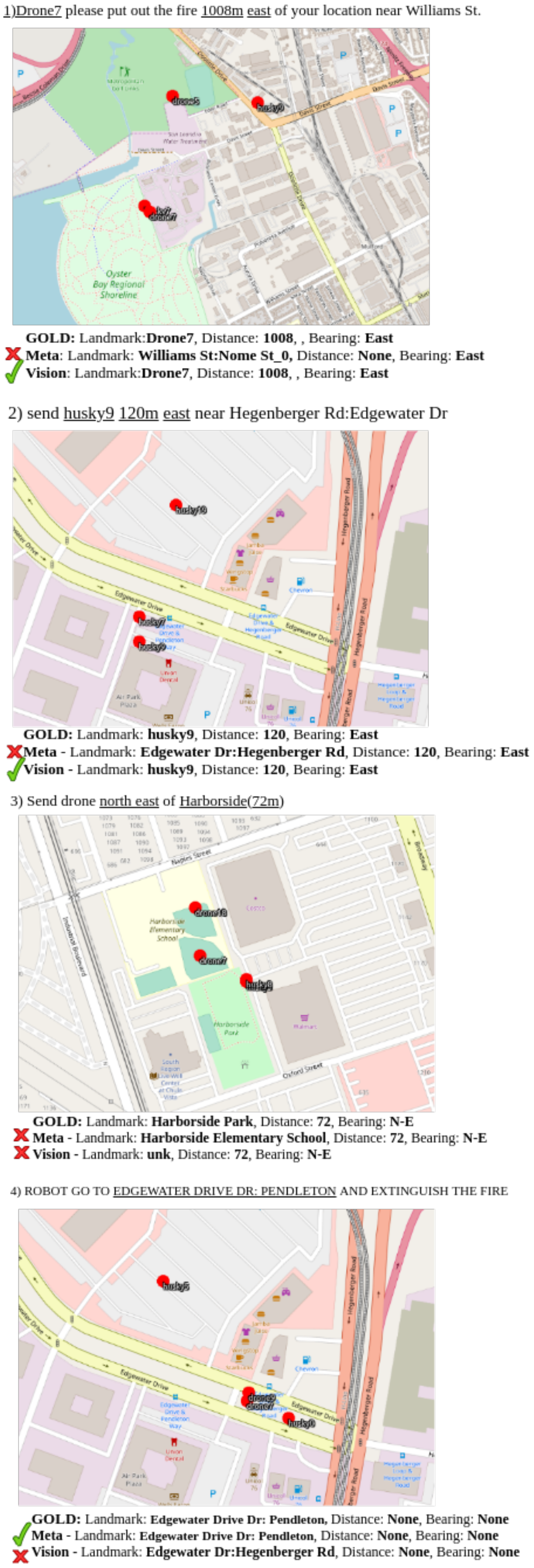
metrics. The first is sized weighted Target error (T err) in meters, which is the distance in meters between the predicted GPS coordinate and the ground truth coordinate. The second is a Pixel Error (P error) which is the difference in pixels between the predicted point in the image and the ground truth converted from the GPS coordinate.

Comparison of Systems We evaluate our system on three variants using different fusion techniques, namely Meta and Language; Meta+Vision and Language; and Vision and Language. Ablations for these systems are shown in Table 1 and are further analyzed in Section 6. We also compare MAPERT to a strong baseline, BERT. The baseline is essentially MAPERT but without the bidirectional cross attention layers in the pipeline (see Fig. 2).

Note, the Oracle of the Meta and Language has a 100% (upper bound) on both cross-validation splits of ROSMI, whereas the oracle of any model that utilizes visual features, is 80% in the 10-fold and 81.98% in the 7-fold cross-validation (lower bound). In other words, the GPS predictor can only work with the output of the automatically predicted entities outputed from Faster R-CNN, of which 20% are inaccurate. Table 1 shows results on both oracles, with the subscript *lower* indicating the lower bound oracle and *upper* indicating the "Upper Bound" oracle. In Table 2, all systems are being projected on the lower bound oracle, so as to compare them on the same footing.

6 Results

Table 2 shows the results of our model for Vision, Meta and Meta+Vision on both the 10-fold cross validation and the 7-fold zero-shot cross validation. We see that the Meta variant of MAPERT outperforms all other variants and our baseline. However, looking at the 10-fold results, Meta+Vision's accuracy of 69.27% comes almost on par with Meta's 71.81%. If we have the harder task of no metadata, with only the visuals of the map to work with, we can see that the Vision component works reasonably well, with an accuracy to 60.36%. This Vision component, despite being on a disadvantage, manages to learn the relationship of visual features with an instruction and vice-versa, compared to our baseline, which has no crossing between the modalities whatsoever, reaching only 33.82%. When we compare these results to the zero-shot paradigm, we see only a 10.5% reduction using Meta, whereas


1)Drone7 please put out the fire 1008m east of your location near Williams St.

GOLD: Landmark:**Drone7**, Distance: **1008**, , Bearing: **East**
✗ **Meta**: Landmark: **Williams St:Nome St_0,** Distance: **None**, Bearing: **East**
✓ **Vision**: Landmark:**Drone7**, Distance: **1008**, , Bearing: **East**

2) send husky9 120m east near Hegenberger Rd:Edgewater Dr

GOLD: Landmark: **husky9**, Distance: **120**, Bearing: **East**
✗**Meta** - Landmark: **Edgewater Dr:Hegenberger Rd**, Distance: **120**, Bearing: **East**
✓**Vision** - Landmark: **husky9**, Distance: **120**, Bearing: **East**

3) Send drone north east of Harborside(72m)

GOLD: Landmark: **Harborside Park**, Distance: **72**, Bearing: **N-E**
✗ **Meta** - Landmark: **Harborside Elementary School**, Distance: **72**, Bearing: **N-E**
✗ **Vision** - Landmark: **unk**, Distance: **72**, Bearing: **N-E**

4) ROBOT GO TO EDGEWATER DRIVE DR: PENDLETON AND EXTINGUISH THE FIRE

GOLD: Landmark: **Edgewater Drive Dr: Pendleton,** Distance: **None**, Bearing: **None**
✓**Meta** - Landmark: **Edgewater Drive Dr: Pendleton**, Distance: **None**, Bearing: **None**
✗**Vision** - Landmark: **Edgewater Dr:Hegenberger Rd**, Distance: **None**, Bearing: **None**

Figure 5: Examples of instructions with the corresponding maps and the accompanied predictions of the best performing either Vision or Meta models conditioned on Oracle$_{lower}$. Underlined words are words corresponding to the target output of the model.

	10-fold Cross Validation (unseen examples)			**7-fold Cross Validation** (unseen scenarios)		
	Accuracy$_{50}$ [SD]	T err [SD]	P err [SD]	Accuracy$_{50}$ [SD]	T err (m)[SD]	P err (m) [SD]
Oracle$_{lower}$	80 [5.01]	23.8 [51.9]	39.1 [96.3]	81.98 [17.09]	20.14 [39]	33.29 [66.43]
Baseline	33.82 [5.16]	64 [57.1]	119.8 [112.3]	34.90 [11.13]	60.71 [57.14]	110.43 [109.71]
Meta	**71.81 [7.37]**	26.70 [47.7]	48.2 [91.2]	**64.30 [14.16]**	32.71 [50.14]	65.71 [88.4]
Vision	60.36 [5.30]	36.40 [51.1]	64.40 [99.6]	49.75 [8.06]	46.00 [54.57]	87.86 [106.0]
Meta+Vision	69.27 [6.68]	26.90 [47.7]	48.30 [91.4]	58.33 [12.24]	36.14 [46.14]	70.71 [93.29]

Table 2: Results on both cross-validations of the best performing ablations of each variant and the baseline. The predictions have been made under the Oracle$_{lower}$. Accuracy (Acc) with IoU of 0.5, Target error (T Err) and Pixel Error (P Err) in meters.

the Vision only component struggles more, with a 17.6% reduction and Vision+Meta a 15.8% reduction. This is understandable since on the 7-fold validation, we tackle unseen maps, which is very challenging for the Vision-only model.

Ablation Study We show ablations for all three model variants in Table 1 and corresponding ablations. We show here just the 10-fold as the 7-fold has similar performance ordering. Depending on the representation of the map for each variant, we derive three ablations for the Meta and two for the Vision. Meta+Vision does not have ablations, since it stands for all possible representations (bb, r, n). Compared to the Oracle$_{lower}$, Meta outperforms the rest, as seen in Table 2. In addition, it requires only the names of the landmarks to score the 71.73%. When we fuse the names and the bboxes, the accuracy decreases slightly, whereas the T err decreases slightly from 26.7 meters to 26.3 meters. The full potential of the Meta model is shown on the Oracle$_{upper}$, which reaches 87.64 % accuracy and T Err of only 9.6 meters, proof that for our task and dataset metadata has the upper hand. It is worthwhile noting that the Vision variant would not have reached 60.36% accuracy, without the r features, since with no fusion of RoI, the accuracy drops to 46.18%.

Error Analysis In order to understand where the Vision and Meta models' comparative strengths lie, we show some example outputs in Fig. 5. In examples 1&2 in this figure, we see the Meta model is failing to identify the correct landmark because the instruction is formulated in a way that allows the identification of two landmarks. It's a matter of which landmark to choose, and the bearing, distance that comes with it, to successfully predict the destination location. However, the Meta model is mixing up the landmarks and the bearings. We believe it is that perhaps the Meta model struggles with spatial relations such as "near". The Vision model, on the other hand, successfully picks up the three correct components for the prediction. This might be helped by the familiarity of the symbolic representation the robots (husky, drones, auvs), which it is able to pick up and use as landmarks in situations of uncertainty such as this one. Both models can fail in situations of both visual and metadata ambiguity. In the third example, the landmark (Harborside Park) is not properly specified and both models fail to pinpoint the correct landmark, since further clarification would be needed. The final example in Fig. 5 shows a situation in which the Meta model works well without the need of a specific distance and bearing. The Vision model manages to capture that, but it fails to identify the correct landmark.

7 Conclusion and Future Work

We have developed a model that is able to process instructions on a map using metadata from rich map resources such as OSM and can do so for maps that it has not seen before with only a 10% reduction in accuracy. If no metadata is available then the model can use Vision, although this is clearly a harder task. Vision does seem to help in examples where there is a level of uncertainty such as with spatial relations or ambiguity between entities. Future work will involve exploring this further by training the model on these type of instructions and on metadata that are scarce and inaccurate. Finally, these instructions will be used in an end-to-end dialog system for remote robot planning, whereby multi-turn interaction can handle ambiguity and ensure reliable and safe destination prediction before instructing remote operations.

Acknowledgments

Katsakioris' PhD was funded by SeeByte, Datalab and MASTS-S. This work was also supported by the EPSRC funded ORCA Hub (EP/R026173/1, 2017-2021).

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Emanuele Bastianelli, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. 2016. A discriminative approach to grounded spoken language understanding in interactive robotics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2747–2753. AAAI Press.

Yonatan Bisk, Kevin Shih, Yejin Choi, and Daniel Marcu. 2018. Learning interpretable spatial operations in a rich 3d blocks world. In *Proceedings of the Thirty-Second Conference on Artificial Intelligence (AAAI-18)*.

Valts Blukis, Nataly Brukhim, Andrew Bennett, Ross A. Knepper, and Yoav Artzi. 2018. Following high-level navigation instructions on a simulated quadcopter with imitation learning. In *Robotics: Science and Systems (RSS)*.

Valts Blukis, Yannick Terme, Eyvind Niklasson, Ross A. Knepper, and Yoav Artzi. 2019. Learning to map natural language instructions to physical quadcopter control using simulated flight. In *Proceedings of the Conference on Robot Learning*.

Johan Boye, Morgan Fredriksson, Jana Götze, Joakim Gustafson, and Jürgen Königsmann. 2014. Walk this way: Spatial grounding for city exploration. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 59–67, New York, NY. Springer New York.

Alexandre de Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. 2015. Artificial neural networks applied to taxi destination prediction. *CoRR*, abs/1508.00021.

Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 165–173, Athens, Greece. Association for Computational Linguistics.

Joyce Yue Chai, Rui Fang, Changsong Liu, and Lanbo She. 2016. Collaborative language grounding toward situated human-robot dialogue. *AI Magazine*, 37:32–45.

David L. Chen and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vision*, 88(2):303–338.

Jana Götze and Johan Boye. 2016. SpaceRef: A corpus of street-level geographic descriptions. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3822–3827, Portorož, Slovenia. European Language Resources Association (ELRA).

Carolin Haas and Stefan Riezler. 2016. A corpus and semantic parser for multilingual natural language querying of OpenStreetMap. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 740–750, San Diego, California. Association for Computational Linguistics.

M. Haklay and P. Weber. 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18.

Stevan Harnad. 1990. The symbol grounding problem. *Phys. D*, 42(1â3):335–346.

M. Hentschel and B. Wagner. 2010. Autonomous robot navigation based on openstreetmap geodata. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, pages 1645–1650.

Baichuan Huang, Deniz Bayazit, Daniel Ullman, Nakul Gopalan, and Stefanie Tellex. 2019. Flight, Camera, Action! Using Natural Language and Mixed Reality to Control a Drone. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019. Tell me more: A dataset of visual scene description sequences. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 152–157, Tokyo, Japan. Association for Computational Linguistics.

J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997.

Miltiadis Marios Katsakioris, Ioannis Konstas, Pierre Yves Mignotte, and Helen Hastie. 2020. Rosmi: A multimodal corpus for map-based instruction-giving. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, pages 680–684, New York, NY, USA. Association for Computing Machinery.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. 2014. Referit game: Referring to objects in photographs of natural scenes. In *Proceedings of EMNLP*.

John D. Kelleher and Geert-Jan M. Kruijff. 2006. Incremental generation of spatial referring expressions in situated dialog. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 1041–1048, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019. Self-monitoring navigation agent via auxiliary progress estimation. *CoRR*, abs/1901.03035.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1475–1482. AAAI Press.

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

Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping instructions to actions in 3D environments with visual goal prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2667–2678, Brussels, Belgium. Association for Computational Linguistics.

Dipendra Misra, John Langford, and Yoav Artzi. 2017. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015, Copenhagen, Denmark. Association for Computational Linguistics.

Teruhisa Misu. 2018. Situated reference resolution using visual saliency and crowdsourcing-based priors for a spoken dialog system within vehicles. *Computer Speech and Language*, 48:1 – 14.

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.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. 2019. Habitat: A platform for embodied AI research. *CoRR*, abs/1904.01201.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2019. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of EMNLP-IJCNLP*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2019. Vision-and-dialog navigation. *CoRR*, abs/1907.04957.

Henry S. Thompson, Anne Anderson, Ellen Gurman Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. 1993. The hcrc map task corpus:

Natural dialogue for speech recognition. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 25–30, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Wenhan Xiong, Xiaoxiao Guo, Mo Yu, Shiyu Chang, Bowen Zhou, and William Yang Wang. 2018. Scheduled policy optimization for natural language communication with intelligent agents. In *Proceedings of IJCAI*, pages 4503–4509.

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2048â2057. JMLR.org.

Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2017. Learning how to learn: An adaptive dialogue agent for incrementally learning visually grounded word meanings. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 10–19, Vancouver, Canada. Association for Computational Linguistics.