A Generative Model for User Simulation in a Spatial Navigation Domain

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
A Generative Model for User Simulation in a Spatial Navigation Domain

Aciel Eshky¹, Ben Allison², Subramanian Ramamoorthy¹, and Mark Steedman¹
¹School of Informatics, University of Edinburgh, UK
²Actual Analytics Ltd., Edinburgh, UK
{a.eshky,s.ramamoorthy,steedman}@ed.ac.uk
ballison@actualanalytics.com

Abstract

We propose the use of a generative model to simulate user behaviour in a novel task-oriented dialog domain, where user goals are spatial routes across artificial landscapes. We show how to derive an efficient feature-based representation of spatial goals, admitting exact inference and generalising to new routes. The use of a generative model allows us to capture a range of plausible behaviour given the same underlying goal. We evaluate intrinsically using held-out probability and perplexity, and find a substantial reduction in uncertainty brought by our spatial representation. We evaluate extrinsically in a human judgement task and find that our model’s behaviour does not differ significantly from the behaviour of real users.

1 Introduction

Automated dialog management is an area of research that has undergone rapid advancement in the last decade. The driving force of this innovation has been the rise of the statistical paradigm for monitoring dialog state, reasoning about the effects of possible dialog moves, and planning future actions (Young et al., 2013). Statistical dialog management treats conversations as Markov Decision Processes, where dialog moves are associated with a utility, estimated online by interacting with a simulated user (Levin et al., 1998; Roy et al., 2000; Singh et al., 2002; Williams and Young, 2007; Henderson and Lemon, 2008). Slot-filling domains have been the subject of most of this research, with the exception of work on troubleshooting domains (Williams, 2007) and relational domains (Lison, 2013).

Although navigational dialogs have received much attention in studies of human conversational behaviour (Anderson et al., 1991; Thompson et al., 1993; Reitter and Moore, 2007), they have not been the subject of statistical dialog management research, and existing systems addressing navigational domains remain largely hand crafted (Janthanam et al., 2013). Navigational domains present an interesting challenge, due to the disparity between the spatial goals and their grounding as utterances. This disparity renders much of the statistical management literature inapplicable. In this paper, we address this deficiency.

We focus on the task of simulating user behaviour, both because of the important role simulators plays in the induction of dialog managers, and because it provides a self-contained means of developing the domain representations which facilitate dialog reasoning. We show how a generative model of user behaviour can be induced from data, alleviating the manual effort typically involved in the development of simulators, and providing an elegant mechanism for reproducing the natural variability observed in human behaviour.

1.1 Spatial Goals of Users

Users in task-oriented domains are goal-directed, with a persistent notion of what they wish to accomplish from the dialog. In slot-filling domains, goals are comprised of a group of categorical entities, represented as slot-value pairs. These entities can be placed directly into the user’s utterance. For example, in a flight booking domain, if a user’s goal is to fly to London from New York on the 3rd of November, then the goal takes the form: \{\text{origin}="\text{New York}"; \text{dest}="\text{London}"; \text{depart date}="03-11-13"\}, and expressing the destination takes the form: \text{Provide dest}="\text{London}".

In contrast, consider the task of navigating somebody across a landscape. Figure 1 shows a pair of maps taken from a spatial navigation domain, the Map Task. Because the Giver aims to communicate their route, one can view the route
### Table 1: A Giver (G) and a Follower (F) alternating turns in a dialog concerning the maps in Figure 1.
The utterances are shown in natural language (left), and the semantic equivalent (right), which is composed of **Dialog Acts** and **Semantic Units**. Utterances marked ⋆ demonstrate a plausible variability in expressing the same part of the route on the Giver’s map, and similarly those marked ⋄. We model the Giver’s behaviour, conditioned on the Follower’s, at the semantic level.

<table>
<thead>
<tr>
<th>Natural Language</th>
<th>Semantic Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>G:</strong> you are above the camera shop</td>
<td>Instruct Position(Above, LM)</td>
</tr>
<tr>
<td><strong>F:</strong> yeah</td>
<td>Acknowledge</td>
</tr>
<tr>
<td><strong>G:</strong> go left jus– just to the side of the paper, * then south, under the parked van ⋄ you have a parked van?</td>
<td>Instruct Move(Towards, Absolute_South) ⋄</td>
</tr>
<tr>
<td><strong>F:</strong> a parked van no</td>
<td>Reply-yn</td>
</tr>
<tr>
<td><strong>G:</strong> you go– you just go west, * and down, and then you go along to the– you go east ⋄</td>
<td>Clarify Move(Towards, Absolute_West) ⋄</td>
</tr>
<tr>
<td><strong>F:</strong> south then east</td>
<td>Check</td>
</tr>
<tr>
<td><strong>G:</strong> yeah</td>
<td>Reply-y</td>
</tr>
</tbody>
</table>

Figure 1: In the Map Task, the instruction Giver’s task is to communicate a route to a Follower, whose map may differ. The route can be seen as the Giver’s goal which the Follower tries to infer. A corresponding dialog is shown in Table 1.

1.2 Utterance Variability for the Same Goal

In addition to making sensible utterances, a concern for user simulation is providing plausible variability in utterances, to provide dialog managers with realistic training scenarios. Consider the dialog in Table 1, resulting from the maps in Figure 1. Utterances marked ⋆ (and similarly those marked ⋄) illustrate how the same route can be described in different ways, not only at the natural language level, but also at the semantic level. A model providing a 1-to-1 mapping from spatial routes to semantic utterances would fail to capture this phenomenon. Instead, we need to be able to account for plausible variability in expressing the underlying spatial route as semantic utterances.

---

627
ations to validate our model. We first use intrinsic
information theoretic measures, which compute
the extent of the reduction in uncertainty brought
by our feature-based representation of the spatial
goals. We then evaluate extrinsically by generating
utterances from our model, and comparing them
to held-out utterance of real humans in the
test data. We also utilise human judgements for
the task, where the judges score the output of the
different models and the human utterances based
on their suitability to a particular route.

2 Related Work

2.1 Related Work on the Map Task

To our knowledge, there are no attempts to model
instruction Givers as users in the Map Task do-
main. Two studies model the Follower, in the con-
text of understanding natural language instructions
and interpreting them by drawing a route (Levit
and Roy, 2007; Vogel and Jurafsky, 2010). Both
studies exclude dialog from their modelling. Al-
though their work is not directly comparable to
ours, they provide a corpus suitable for our task.

2.2 Related Work on User Simulation

Early user simulation techniques are based on N-
grams (Eckert et al., 1997; Levin and Pieraccini,
2000; Georgila et al., 2005; Georgila et al., 2006),
ensuring that simulator responses to a machine ut-
terance are sensible locally. However, they do not
enforce user consistency throughout the dialog.

Deterministic simulators with trainable parame-
ters mitigate the lack of consistency using rules in
conjunction with explicit goals or agendas (Schef-
fler and Young, 2002; Rieser and Lemon, 2006;
Pietquin, 2006; Ai and Litman, 2007; Schatzmann
and Young, 2009). However, they require large
amounts of hand crafting and restrict the variability
in user responses, which by extension restricts
the access of the dialog manager to potentially in-
teresting states. An alternative approach to dealing
with the lack of consistency is to extend N-grams
to explicitly model user goals and condition utter-
ances on them (Pietquin, 2004; Cuayáhuhtil et al.,
2005; Pietquin and Dutoit, 2006; Rossignol et al.,
2010; Rossignol et al., 2011; Eshky et al., 2012).

3 The Model

Our task is to model the Giver’s utterances in re-
response to the Follower’s, at the semantic level. A
Giver’s utterance takes the form:

\[ g = \text{Instruct}, \quad u = \text{MOVE (UNDER, LM)} \]

consisting of a dialog act \( g \) and a semantic unit \( u \).
Aligned with \( u \), is an ordered set of waypoints \( W \),
corresponding only to part of the route \( u \) describes.
Figure 2(a) shows an example of such a sub-route.
The point-set \( W \) can be seen as the Giver’s current
goal on which they base their behaviour. Because
the routes are drawn on the Giver’s maps, we treat
\( W \) as observed.

To model some of the interaction between the
Giver and the Follower, we additionally consider
in our model the previous dialog act of the Fol-
lower, which could for example be:

\[ f = \text{Acknowledge} \]

Given point-set \( W \) and preceding Follower act
\( f \), as the giver, we need to determine a procedure
for choosing which dialog act \( g \) and semantic unit
\( u \) to produce. In other words, we are interested in
the following distribution:

\[
p(g, u|f, W) \quad (1)
\]

which says that, as the Giver, we select our utter-
ances on the basis of what the Follower says, and
on the set of waypoints we next wish to describe.

To formalise this idea into a generative model,
we assume that the Giver act \( g \) depends only on
the Follower act \( f \). We further assume that the se-
mantic unit \( u \) depends on the set of waypoints \( W \)
which it describes, and on the Giver’s choice of
dialog act \( g \). Thus, \( u \) and \( f \) are conditionally in-
dependent given \( g \). This provides a simple way of
incorporating the different sources of information
into a complete generative model\(^3\). Using Bayes’
theorem, we can rewrite Equation (1) as:

\[
p(g, u|f, W) = \frac{p(u) p(g|u) p(f|g) p(W|u)}{\sum_{u’} p(u’)} \quad (2)
\]

requiring four distributions: \( p(u) \), \( p(g|u) \), \( p(f|g) \), and
\( p(W|u) \). The first three become the seman-
tic component of our model, to which we dedi-
icate Section 3.1. The fourth is the spatio-semantic
component, to which we dedicate Sections 3.2–
3.4.

\(^3\)We align \( g \) and \( u \) in a preprocessing step, and store the
names of landmarks which the units abstract away from.

Further advancements to this work would investigate the
effects of relaxing the conditional independence assumption.
3.1 The Semantic Component
The semantic component concerns only the categorical variables, f, g, and u, and addresses how the Giver selects their semantic utterances based on what the Follower says. We model the distributions u, g|u, and f|g from Equation (2) as categorical distributions with uniform Dirichlet priors:

\[ u \sim \text{Cat}(\alpha) \quad \alpha \sim \text{Dir}(\epsilon) \quad (3a) \]
\[ g|u \sim \text{Cat}(\beta) \quad \beta \sim \text{Dir}(\kappa) \quad (3b) \]
\[ f|g \sim \text{Cat}(\gamma) \quad \gamma \sim \text{Dir}(\lambda) \quad (3c) \]

We use point estimates for \( \alpha, \beta \) and \( \gamma \), fixing them at their posterior means in the following manner:

\[ \hat{\beta}_{gu} = p(g|u) = \frac{\text{Count}(g,u) + 1}{\sum_{g'} \text{Count}(g',u) + L} \quad (4) \]

and similarly for \( \hat{\alpha} \) and \( \hat{\gamma} \) \( (L = \text{size of vector } \beta) \).

3.2 Spatial Goal Abstraction
Each ordered point-set \( W \) on some given map can be seen as the Giver’s current goal, on which they base their behaviour. Let \( W = \{w_i; 0 \leq i < n\} \), where \( w_i = (x_i, y_i) \) is a waypoint, and \( x_i, y_i \) are pixel coordinates on the map, typically obtained through a vision processing step.

Given this goal formulation, from Equation (2) we require \( p(W|u) \), i.e. the probability of a set of waypoints given a semantic unit. However, there are two problems with deriving a generative model directly over \( W \). Firstly, the length of \( W \) varies from one point-set to the next, making it hard to compare probabilities with different numbers of observations. Secondly, deriving a model directly over \( x, y \) coordinates introduces sparsity problems, as we are highly unlikely to encounter the same set of coordinates multiple times. We thus require an abstraction away from the space of pixel coordinates.

Our approach is to extract feature vectors of fixed length from the point-sets, and then derive a generative model over the feature vectors instead of the point-sets. Feature extraction allows point-sets with similar characteristics, rather than exact pixel values, to give rise to similar distributions over units, thus enabling the model to reason given previously unseen point-sets. The features we extract are detailed in Section 3.4.

3.3 The Multivariate Normal Distribution
Let \( M \) be an unordered point-set describing map elements, such as landmark locations and map boundary information. \( M = \{m_j; 0 \leq j < k\} \), where \( m_j = (x_j, y_j) \) is a map element with pixel coordinates \( x_j \) and \( y_j \). We define a spatial feature function \( \psi : W, M \rightarrow \mathbb{R}^n \) which captures, as feature values, the characteristics of the point-set \( W \) in relation to elements in \( M \). Let the spatial feature vector, extracted from the point-set \( W \) and the map elements \( M \), be:

\[ v = \psi(W, M) \quad (5) \]

Figure 2(b) illustrates the feature extraction process. We now define a distribution over the feature vector \( v \) given the semantic unit \( u \). We model \( v|u \) as a multivariate normal distribution (recall that \( v \) is in \( \mathbb{R}^n \)):

\[ v|u \sim N(\mu_u, \Sigma_u) \quad (6) \]

where \( \mu \) and \( \Sigma \) are the mean vectors and covariance matrices respectively. Subscript \( u \) indicates that there is one such parameter for each unit \( u \).

Since the alignments between units \( u \) and point-sets \( W \) are fully observed, parameter estimation is a question of estimating the mean vectors \( \mu_u \) and the covariance matrices \( \Sigma_u \) from the point-sets co-occurring with unit \( u' \). We use maximum likelihood estimators. To avoid issues with degenerate covariance matrices resulting from small amounts of data, we consider diagonal covariance matrices. Because \( v|u \) is normally distributed, inference, both for parameters and conditional distributions over units, can be performed exactly, and so the model is exceptionally quick to learn and perform inference.

3.4 The Spatial Feature Sets
We derive four feature sets from the ordered point-set \( W \), while considering the map elements in the unordered point-set \( M \):

1. **Absolute features** capture directions and distances of movement. We compute the distance between the first and last points in \( W \), and compute the angle between unit vector \(<0,-1> \) and the line connecting first and last points in \( W \).

2. **Polynomial features** capture shapes of movements as straight lines or curves. We compute the mean residual of a degree one polynomial fit to the points in \( W \) (linear), and a degree two polynomial (quadratic)\(^4\)

\(^4\)These features are computed quickly and efficiently, requiring only the solution to a least squares problem.
3. **Landmark features** capture how close the route takes the Follower to the nearest landmark. We compute the distance between the end-point in $W$ and the nearest landmark in $M$, and compute the angle between the route taken in $W$ and the line connecting the start point in to the nearest landmark.

4. **Edge features** capture the relationship between the movement and the map edges. We compute the distance from the start-point in $W$ to the nearest edge and corner in $M$, and similarly for the end-point in $W$.

### 3.5 The Complete Generative Model

Our complete generative model of the Giver is a distribution over Giver act $g$ and semantic unit $u$, given the preceding Follower act $f$, and the spatial feature vector $v$. Vector $v$ is the result of applying the feature extraction function $\psi$ over $W$ and $M$, where $W$ is the ordered point-set describing the sub-route aligned with $u$, and $M$ is the point-set describing landmark locations and map edge information. We rewrite Equation (2) as:

$$p(g, u|f, v) = \frac{p(u) p(g|u) p(f|g) p(v|u)}{\sum_{u'} p(u) p(g'|u) p(f'|g') p(v|u')} \tag{7}$$

We call our model the Spatio-Semantic Model, SSM, and depict it in Figure 2(c).

### 4 Corpus Statistics and State Space

We conduct our experiments on the Map Task corpus (Anderson et al., 1991), a collection of cooperative human-human dialogs arising from the task explained in Figure 1 and Table 1. The original corpus was labelled with dialog acts, such as Acknowledge and Instruct. The semantic units can be obtained through a semantic parse of the natural language utterances, while the spatial information can be obtained through vision processing of the maps. We use an existing extension of the corpus by Levit and Roy (2007), which is semantically and spatially annotated. The spatial annotation are $x, y$ pixel coordinates of landmark locations and evenly spaced points on the routes. All 15 maps were annotated. The semantic units take the predicates Move, Turn, Position, or Orientation, and two arguments: a route descriptor and a referent. The semantic annotations were restricted to the Giver’s Instruct, Clarify, and Explain acts. Out of the original 128 dialogs, 25 were semantically annotated.

For our experiments, we use all 15 pairs of maps, and all 25 semantically annotated dialogs. A dialog on average contain 57.5 instances, where an instance is an occurrence of $f, g, u,$ and $W$. We find 87 unique semantic units $u$ in our data, however, according to the semantic representation, there can be 456 distinct possible values for $u^5$. As for the rest of the variables, $f$ takes 15 values, $g$ takes 4, and $v$ is a real-valued vector of length 10, extracted from the real valued sets $W$ and $M$ of varying lengths. We thus reason in a semantic state space of $87 \times 15 \times 4 = 5220$, and an infinite spatial state space.

### 5 Intrinsic Evaluation

Our first evaluation metric is an information theoretic one, based on notion that better models find new instances of data (not used to train them) to be more predictable. One such metric is the probability a model assigns to the data, (higher is better). A

$$20 \times 2$$ for Turn and Orientation, $+ 208 \times 2$ for Move and Position.
second metric is perplexity, which computes how surprising a model finds the data (lower is better). Both metrics have been used to evaluate user simulators in the literature (Georgila et al., 2005; Eshky et al., 2012; Pietquin and Hastie, 2013). We compute the per-utterance probability of held-out data, instead of the per-dialog probability, since the latter was deemed incompatible across dialogs of different lengths by Pietquin and Hastie (2013). Perplexity is \(2^{-\log_2(d)}\) where \(d\) is the probability of the instance in question. We evaluate using leave-one-out validation, which estimates the model from all but one dialog, then evaluates the probability of that dialog. We repeat this process until all dialogs have been evaluated as the unseen dialog.

Because we evaluate on held-out dialogs, we need to be able to assign probabilities to previously unseen instances. We therefore smooth our models (at training time) by learning a background model which we estimate from all the training data. This results in high variance in the distribution over features and a flat overall distribution. Where no model can be estimated for a particular semantic unit, we use that semantic unit’s smoothed prior probability combined with the background model for its likelihood.

We first consider the suitability of the different feature sets for predicting utterances. Figure 4 shows the mean per-utterance probability our model assigns to held-out data when using different sets. The more predictable the model finds the data, the higher the probability. Note that the target metric here is not \(1\), as there is no single correct answer. It can be seen that the most successful features in order of predictiveness are: Absolute, then Polynomial, then Landmark, and finally Edge. The combination of all buys us further improvement. Perplexity is shown in Table 2.

![Image](image_url)

**Figure 3:** Mean per-utterance probability, assigned to held-out data by our model, when defined over the four feature sets and their combinations, estimated through leave-one-out validation. A=Absolute, P=Polynomial, L=Landmark, and E=Edge. Error bars are standard deviations.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute (A)</td>
<td>7.26 ± 4.08</td>
</tr>
<tr>
<td>Polynomial (P)</td>
<td>12.86 ± 8.39</td>
</tr>
<tr>
<td>Landmark (L)</td>
<td>15.16 ± 6.27</td>
</tr>
<tr>
<td>Edge (E)</td>
<td>17.92 ± 8.47</td>
</tr>
<tr>
<td>All</td>
<td>4.66 ± 2.22</td>
</tr>
</tbody>
</table>

**Table 2:** Perplexity scores (and standard deviations) of our model, computed over the four feature sets and their combinations, estimated through leave-one-out validation. (A) outperforms all individual sets, while the combination performs best.

Secondly, we consider two baselines inspired by similar approaches of comparison in the literature (Eckert et al., 1997; Levin and Pieraccini, 2000; Georgila et al., 2005). Both are variants of our model that lack the spatial component, i.e. they are not goal-based. Although the baselines are weak, they allow us to measure the reduction in uncertainty brought by the introduction of the spatial component to our model, which is the purpose of this comparison. **Baseline 1** is \(p(g, u)\) while **Baseline 2** is \((g, u|f)\). The first tells us how predictable giver utterances are (in the held-out data), based only on the normalised frequencies. The second tells us how predictable they become when we condition on the previous follower act. Details of the baselines are similar to Section 3.1.

Figure 4 shows the mean per-utterance probability our model assigns to held-out data when compared to the two baselines. Baseline 2 slightly improves our predictions over Baseline 1, although not reliably so, when considering the small increase in perplexity in Table 3. SSM demonstrates a much larger relative improvement across both metrics. The results demonstrate that our spatial component enables substantial reduction in uncertainty, brought by the transfer of information from the maps to the utterances.

Intrinsic metrics, such as the probability of held-out data and perplexity, provide us with an elegant way of evaluating probabilistic models in a setting where there is no single correct answer, but a range of plausible answers, because they exploit the model’s inherit ability to assign probability to behaviour. However, the metrics can be hard to interpret in an absolute sense, providing much better
information about the relative strengths of different models rather than their absolute utility. In the next section, we explore methods for determining the utility of the models when applied to tasks.

6 Extrinsic Evaluation

In this section, we undertake a task-based evaluation of model output. We train on 22 of the dialogs, holding out 3 at random for testing. The task is to then generate, for each sub-route in the test dialogs, the most probable unit to describe it. Figure 5 shows some examples of sub-routes taken from the test dialogs, and shows the most probable unit to describe each under our model, SSM.

We first explore a naive notion of accuracy: the percentage of model-generated units matching Real Giver units observed in the test dialogs. We compute the same for Baseline 1 from Section 5 as a lower bound. A quick glance at the results in Table 4 might suggest that both models have little utility: SSM is “correct” only 33% of the time. However, the extent to which this conclusion follows depends on the suitability of accuracy as a means of evaluating dialog. In most situations, there is not a single correct description and a host of incorrect ones, but rather a gradient of descriptions from the highly informative and appropriate to the nonsensical and confusing. Such subtleties are not captured by an accuracy test (or the closely related recall and precision). In demonstration of this point, we next conduct qualitative evaluation of model output.

We ask humans to rate, on a Likert scale of 7, the degree to which a given unit provides a suitable description of a given sub-route. Sub-routes are taken from the test dialogs, and are marked similarly to Figure 5 but on the complete map. Units are generated from SSM, Baseline, Real Giver, and a control condition: a deliberate Mismatch to the sub-route. The Mismatch is generated automatically by taking the least probable unit under SSM, of the form MOVE(TOWARD, x) where x is one of the four compass directions. We collect 5 judgements for each sub-route-unit combinations on Mechanical Turk, and randomise so that no judge sees the same order of pairs. Test dialogs contained 94 distinct sub-routes.

We analyse the results with a two-way ANOVA, with the first factor being model, and the second being the sub-route, for a $4 \times 94$ design. The means of the “model” factor are shown in Table 5. It can be seen that Mismatch and Baseline are scored sensibly poorly, while SSM and Real Giver are scored reasonably well, and are judged to be of a similar quality. We thus proceed with a more rigorous analysis. The ANOVA summary is shown in Table 6. A significant effect of the model fac-
Figure 5: Given a sub-route marked with start-point ◦ and end-point × (in red), SSM generates the following u: (a) MOVE(TOWARDS, ABS_NORTHEAST) (b) TURN(ABS_WEST) (c) MOVE(FOLLOW-BOUNDARY, LM) (d) MOVE(AROUND, LM) (e) MOVE(TOWARDS, ABS_SOUTHWEST)

Table 6: Two way ANOVA with factors model (4 possibilities), and sub-route (94 possibilities). Results show a model effect accounting for most of the variance. Meaning that the scores assigned to the units by human judges are significantly influenced by the model used to generate the units.

Table 7: Tukey HSD test, demonstrating that all models are significantly different from one another, except Real Giver and SSM. Results show that, despite the large number of judgements collected, we are unable to separate the quality of our model’s unit from that in the original data, against which accuracy was being judged in Table 4. This demonstrates that when many answers are feasible, scoring correctness against the original human units is unsuitable. It also firmly demonstrates the suitability of our spatial representation, and the strength of the generative model we have induced for the task.

7 Conclusion and Discussion

We have shown how to represent spatial goals in a navigational domain, and have validated our representation by inducing (fully from data) a generative model of the Giver’s semantic utterances conditioned on the spatial goal and the previous Follower act. Intrinsic and extrinsic evaluation demonstrate the strength of our model.

A direct application of this work is robot guidance, by using the Giver’s simulator to induce an optimal Follower: an MDP-based dialog manager that interprets and follows navigational instructions. Another variation would be to learn a generative model of the Follower, by extracting features from Follower maps (labelled with routes drawn by real Followers). Finally, this work has broader applications beyond simulation, in particular for systems that describe routes to users (spatial goal representation and model dependencies would hold). Decisions about which part of the route to describe next is one extension to that end.

Acknowledgements

We thank Ioannis Konstas, Johanna Moore, Robin Hill, S. M. Ali Eslami, and the anonymous reviewers for valuable feedback. This work is funded by King Saud University. Mark Steedman is supported by EC-PF7-270273 Xperience and ERC Advanced Fellowship 249520 GRAMPLUS.
References


Srinivasan Janarthanam, Oliver Lemon, Phil Bartie, Tiphaine Dalmas, Anna Dickinson, Xingkun Liu, William Mackaness, and Bonnie Webber. 2013. Evaluating a city exploration dialogue system with integrated question-answering and pedestrian navigation. In ACL.


