Building and Applying Perceptually-Grounded Representations of Multimodal Scene Descriptions

Ting Han   **Casey Kennington**   **David Schlangen**
CITEC / Dialogue Systems Group / Bielefeld University
`first.last@uni-bielefeld.de`

Abstract

"You see a red building, and then behind that [gesture] you turn left". Hearing this kind of route description, only to apply its instructions at a later time, is a difficult task. The content of the description has to be memorised, and then, when the time comes to make use of it, be applied to the present situation. This makes for a good test case for a model of situated dialogue understanding, as the accuracy and applicability of a constructed multimodal content representation can be directly tested. In this paper, we present a model of a simplified version of this general task (namely, describing spatial scenes) and discuss three variants for realising the 'extraction', memorisation, and application of the content of route descriptions. We evaluate the approach and the variants with an implementation of the model, using a corpus of descriptions and application situations.

1 Introduction

Describing routes to destinations not currently in view, then understanding and later following such descriptions, is among the hardest language-related tasks (Schneider and Taylor, 1999). The description giver must imagine to herself the spatial layout of the scene through which the route leads, must imagine movement through that scene, and then encode all this in speech and gestures. The recipient in turn must represent to himself the content of the description, in such a way that it can later indeed form the basis for navigation.

In this paper, we model the task of the *recipient* of such a description, in a somewhat simplified version. Being presented with a multimodal description of a spatial configuration of landmarks,

such as illustrated by (1) and gestures as shown in Figure 1 below, we assume that the recipient builds a representation of the content of this description and is then later able to use this representation to recognise the described scene in a set of candidate scenes.

(1)   There is a red circle here [gesture] and slightly above it [gesture] a blue L.

We propose, compare and explore a range of different models for building and applying such representations, which we implement in a dialogue processing system and evaluate on a corpus of scene descriptions.

Figure 1: Providing a multimodal scene description

Our desiderata for these representations are as follows: That they represent equally the contribution by language and by gesture (which goes beyond what formal semantics-based approaches typically do); that the mapping from lexical entries to non-logical constants is well-motivated (where in contrast in formal semantics what should be arbitrary symbols is often suggestively named, e.g. `red` as translation of the word *red*); and that these non-logical constants are perceptually grounded (and not just equiped with a model-theoretic in-

terpretation that simply states the extension).

We describe in general terms the structure of the representations in the following section, along with the variants within that structure that we explore. In Section 3 we further specify the modelling task and describe how we implemented it. The implementation is then evaluated in Section 4. We then discuss related literature and conclude.

2 Representing Scene Descriptions

Descriptions such as illustrated by example (1) form a type of mini-discourse, where referents for objects are introduced into the discourse and constraints are added. The basic structure of the representation format we use is consequently inspired by Discourse Representation Theory (Kamp and Reyle, 1993), as can be seen by the representation (schema) for the example given in (2).

(2)
$$\boxed{\begin{array}{l} o_1, g_1, o_2, g_2 \\ \hline o_1\colon transl(\text{red circle}) \\ g_1 : (x_1, y_1) \\ pos(o_1, \phi(g_1)) \\ slightly\_above(o_1, o_2) \\ o_2\colon transl(\text{blue L}) \\ g_2 : (x_2, y_2) \\ pos(o_2, \phi(g_2)) \end{array}}$$

The gestural component is represented by the "gesture referents" g_1 and g_2, and we simplify in assuming that they only contribute a single point in space (the position of the stroke of the deictic gesture). The connection between the verbal content and the gestural content is indicated by predicates stating that the gestures, respectively, specifiy the positions of the objects.[1]

We explore three different ways of filling in the details that are glossed over in (2) with the function $transl()$, which is supposed to translate from the utterances to its logical form.

- In **Variant A**, we translate the referring expressions simply into a sequence of lemmata. This would lead to a representation of "red circle" as $red, circle$.

- In **Variant B**, the translation proceeds by specifying a semantic frame (Fillmore,

---

[1] Following Lascarides and Stone (2009), we assume that they do this via a context-specific function that maps the positions in gesture space to the intended real-world positions; but we do not further develop this part here.

1982), but here by way of more practically-oriented approaches to spoken language understanding (Tur and De Mori, 2011)) for object descriptions, leading to, for example $\begin{bmatrix} shape : circle' \\ colour : red' \end{bmatrix}$ for "red circle". This presupposes availability of a process that can do such a mapping; e.g., a lexicon that links lexical items and such frame elements, and a pre-specified repertoire of attributes and values for them.

- In **Variant C** finally, we map the referring expression into a sequence of symbols (similar to Variant B) where however the repertoire of these symbols comes from an automatic learning process, and thus does not necessarily correspond to pre-theoretic notions of the meaning of such attributes.

All variants have in common that the symbols used in the representation are perceptually grounded, that is, their applicability in a given context can be determined by representing that context through perceptual (here, visual) features.

To make these proposals more concrete, we put them to use in a specific application, which will be described in the next section.

3 Processing Scene Descriptions

3.1 The Scene Retrieval Task

The specific task that we are modelling is the following: Given – in real time, word by word – a verbal/gestural description as in (1), construct a representation of the relevant content. Then, when the representation is built, use it to identify in a set of visually presented scenes the one that best conforms to the description. The task hence requires a) *constructing* the representation, based on perceived speech and gestures, and b) *applying* it in a (later) visually perceived context. Performance on the retrieval task gives a practical measure for the quality of the representation; if the representation does indeed capture the relevant content, it should form the basis for identifying that what was described.

Figure 2 shows an example scene which we used in our evaluation experiments. In all of the scenes, there are three puzzle pieces (more precisely, pentomino pieces constructed out of 5 squares in different configurations, which leads to 12 possible shapes, from which three are randomly
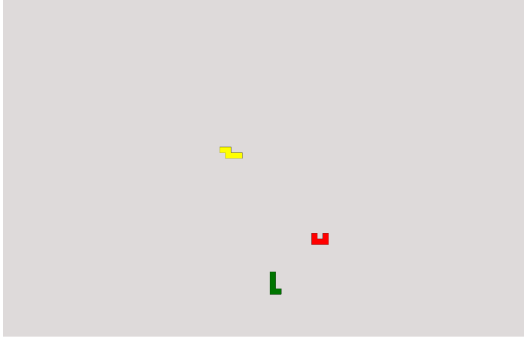
Figure 2: Scene example

selected), with a randomly determined color and position.

3.2 The Processing Pipeline

The verbal/gestural description of a scene is processed by a processing pipeline as illustrated in Figure 4. As shown in the figure, the system takes speech and gesture as input. Speech is processed by an ASR which produces output word-by-word. The output then is fed into a segmentation module that decides when a new object is introduced in the discourse. In parallel, a motion capture sensor records hand motion data and sends the data to a deictic gesture detector. This detector sends a signal to the segmentation module when a deictic gesture is detected. The segmentation module integrates the deictic gesture with the corresponding object information. The integrated information is then sent to a representation module which builds the representation for the incoming description.

At a later time, and after the full description has been perceived, it is used to make a decision in the retrieval task. The scenes among which the described one is to be found are given (as images) to a computer vision module, which recognises the objects in the scenes and computes a feature vector for each, containing information about the colour of the object, the number of edges, its skewness, position, etc.; i.e., crucially, the object is not represented by a collection of symbolic property labels, but by real-valued features. The application module takes this representation for each candidate scene as input, and computes a score for how well the stored representation of the description content matches the candidate scene. For this, it makes use of the perceptually-grounded nature of the symbols used in the content representation, which connect these with the object feature vectors. The scene with the highest score finally is

chosen as the one that is retrieved.

In the following, we describe some of these processing steps in more detail.

3.3 Applying Gestural Information

As described above, the discourse representation includes information about positions indicated by the gestures. To make use of this information in distinguishing between scenes, the first step is to compute for each scene the likelihood that it, with the position that objects are in, gave rise to the observed (and represented) gesture positions. This is not as trivial as it may sound, as the gesture positions are represented in a coordinate system given by the motion capture system, whereas the object positions are relative to the image coordinate system. Moreover, the gestures may have been performed sloppily. Finally, on a more technical level, the labels that the segmentation module assigned to the parts of the description (o_a etc. in Figure 4) do not immediately map to those given to the objects recognised by the computer vision module (o_1 etc.).
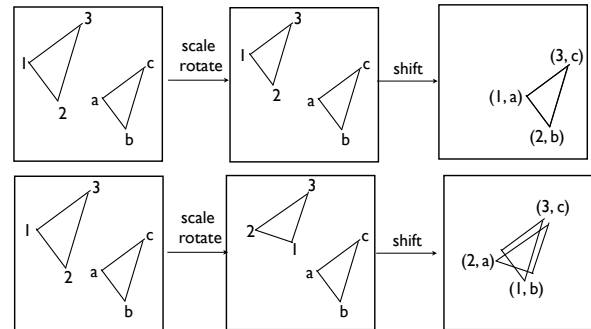

Figure 3: Example of a good mapping (top) and bad mapping (bottom), numbered IDs represent the perceived objects, the letter IDs represent the described objects.

To address the latter question (which description objects to compare with which computer vision object), we simply try all permutations of mappings. For each mapping a score is then computed for how well the gestured configuration under a given mapping can be transformed into the scene configuration. This is illustrated in Figure 3. First, the gestured configuration is projected into the same coordinate system as the scene configuration, and then it is scaled, rotated and shifted to be as congruent with the scene configuration as possible. In Figure 3, where the top target mapping between description object IDs and scene ob-
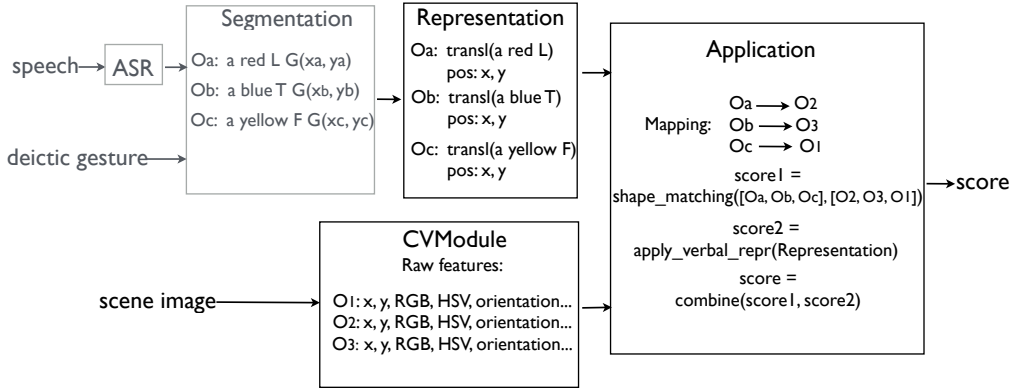
Figure 4: Processing pipeline

ject IDs is sensible, this operation leads to a good fit, the bottom mapping is not as good. (It will be even worse when attempting to map a gesture configuration into a scene configuration that is wildly different; e.g. a triangle into a sequence of objects placed in one line.) We assume that we have available a model trained on observed gestures for known positions, which can turn this distance score into a probability (i.e., the likelihood of this gesture configuration being observed when the scene configuration is the intended one).

On a technical level, this works as follows. The positions of the three objects in the description and in the visual scene can be represented as matrices S_d and S_v of the form:

$$S = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{pmatrix} \quad (1)$$

With a set of parameters p

$$p = [\theta, t_x, t_y, s] \quad (2)$$

where θ is the rotating angle; t_x (t_y) stands for the shift value on the x (y) axis; s is the scaling parameter. We scale, rotate and shift matrix S_v to get a transformed matrix S_t:

$$S_t(x,y) = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3)$$

By minimizing the cost function:

$$E = min \parallel S_t - S_d \parallel \quad (4)$$

we compute the optimal p. The distance between the resulting optimal S_t and S_d gives a metric for the goodness of the result, which is the input for a likelihood model that turns this into a probability.

3.4 Knowledge from Prior Experience

We assume that our system brings with it knowledge from previous experience with object descriptions. This knowledge is used (at least in some variants) for the task of mapping to logical form, and in all variants for the perceptual grounding of the symbols in the logical form. In what follows, we first briefly describe the corpus of interactions from which this prior knowledge is distilled.
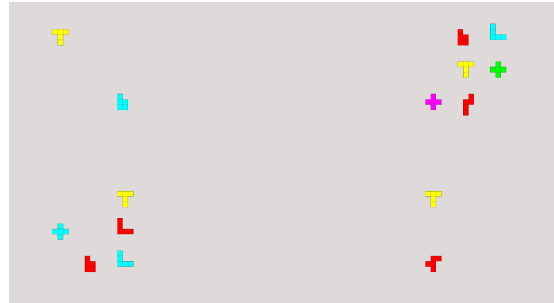
3.4.1 The corpus: TAKE

Figure 5: Example TAKE scene used for training.

As source for this knowledge, we use the TAKE corpus (Kousidis et al., 2013). In a Wizard-of-Oz study, participants were presented on a computer screen with a scene of pentomino pieces (as in Figure 5) and asked to identify one piece to a "system" by describing and pointing to it. The utterances, arm movements, scene states and gaze information were recorded as descibed in Kousidis et al. (2012). In total, 1214 episodes were recorded from 8 participants (all university students). The corpus was further processed to include raw visual features (such as color, shape HSV, RGB values etc.) of each pento tile for each scene (in

	DESCRIPTION	REPRESENTATION	APPLICATION	VISUAL INPUT
A		word stems	word classifiers	
B	speech + gesture	property labels	property classifiers	raw visual object and scene features
C		cluster labels	cluster classifiers	

Table 1: Overview of variants.

the same way as the computer vision module described above does); it also includes for each object symbolic properties (e.g., *green*, *X* (a shape)), and for the intended referent the utterance that the participant used to refer to it.

3.4.2 (Learning) Mappings to Logical Form

As described above, one difference between the variants of our model lies in how they realise the $transl()$ function from representation (2). Only variant B actually uses the data to learn this mapping, but we describe all variants here. In all variants, there is a preprocessing step that normalises word forms (by *stemming* them using NLTK (Loper and Bird, 2002)). This will map for example all of *grün, grünes, grüne* into *grun*. Thus, this step reduces the size of the vocabulary that needs to be mapped.

Variant A For variant A, stemming is all that is done in terms of mapping into logical form, and an object description is translated into the sequence of its stemmed words.

Variant B For variant B, similar to the model presented in (Kennington et al., 2013), we learn a simple mapping from words to symbolic property labels, based on co-occurrence in the training data. (E.g., we will have observed that the word *green* occured when the referent had the property green, strengthening the link between that word and that property.) The model gives us for each word a probability distribution over properties; we chose the most likely property (averaging over the contribution of all words) as the representation for the description. Note that this variant does not require a pre-specified lexicon linking words to properties, but it does require a pre-specified set of properties (e.g., green, red, etc., totaling 7 colour and 12 shape properties).

Variant C We overcome the latter limitations (pre-specifying properties) in this variant. As will be described below, for variant A we learn for each word (stem) a classifier that links it to perceptual input. These classifiers themselves can be represented as vectors (the regression weights of the logistic regression, see below). Using the intuition

that words with similar meaning should give rise to similarly behaving classifiers (e.g., the classifier to "light green" should respond similarly–not identically–to that for "green"), we ran a clustering algorithm (k-means clustering) on the set of classifier vectors. The resulting clusters, through their centroids, can then themselves be turned again into classifiers. This effectively reduces the number of classifiers that need to be kept, just as in variant B the set of properties is smaller than the set of words that are mapped into it, but here the clusters are chosen based on the data, and not on prior assumptions. The object description then is represented as a sequence of the labels of those clusters that the words in the description map into.

Table 1 shows an overview of the variants; their input and representation which make up how the descriptions are compressed and stored, and application and visual input which comprises how the scenes are perceived and applied.

3.4.3 Learning Perceptual Groundings

Variant A For variant A, we learned grounded word (stem) meanings in a similar way as done in Kennington et al. (2015): For each word stem w occurring in the TAKE corpus of referring expressions, we train a binary logistic regression classifier (see (5) below, where $\mathbf{w}$ is the weight vector that is learned and σ is the logistic function) that takes a visual feature representation of a candidate object ($\mathbf{x}$) and is asked to return a probability p_w for this object being a good fit to the word. We present the object that the utterance referred to as a positive training example for a good fit, and objects that it didn't refer to as a negative example. (See Kennington et al. (2015) for a discussion of the merits of this strategy.)

$$p_w(\mathbf{x}) = \sigma(\mathbf{w}^\mathsf{T}\mathbf{x} + b) \qquad (5)$$

As mentioned above, each classifier is fully specified by its coefficients ($\mathbf{w}$ and b).

Variant B As described above, the first step in variant B was to use the words in the object description as evidence for how to fill the semantic
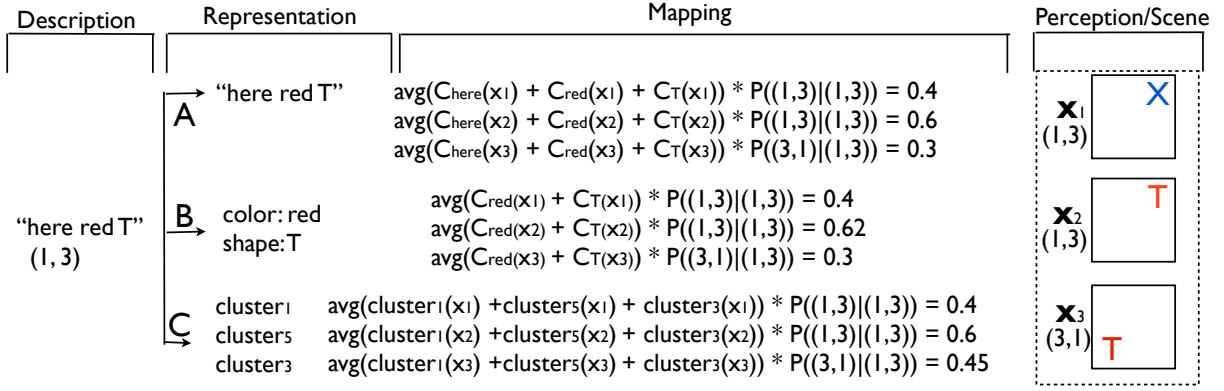
Figure 6: Simplified (and constructed) pipeline example. The description "here a red T" with gesture at point (1,3) is represented and mapped to the perceived scenes. Each variant assigns a higher probability to the correct scene, represented by X_2

frame, with the frame elements *colour* and *shape*. For the possible values of these elements (e.g., `green`) we trained the same type of logistic regression classifier, again using cases where the property was present for a given object as positive example and, as negative examples, those where it wasn't. This then gives a perceptual grounding for the property `green` (whereas in variant A we trained one for the *word* "green"). In a way, this variant begs the question of where the ontology of properties comes from; if this part is a model of language acquistion, the claim would be that there is a set of innate labels which just need to be instantiated.

Variant C As described above, this variant builds on variant A, by reducing the set of classifiers that are required through clustering. In the experiment described below, we set the number of clusters to compute to 26 (an experimentally determined optimum), which resulted for example in one cluster grouping together "violett" and "lila" (*violet* and *purple*), or another one clustering "türkis, blau, dunkelblau" (*turquoise, blue, dark blue*), but also clusters that are less readily interpreted such as "nochmal, rosa, hmm" (*again, pink, erm*). What is important to note here in any case is that the reduction in the range of what words can map into in their semantic representation is as strong as with B, but emerges from the data.

3.4.4 Applying the Information

With all this in hand, the final score for a given candidate scene is computed as follows. For each possible mapping of description object IDs into computer vision object IDs, a gestural score is computed as described in Section 3.3; the representation of each description is applied to its corresponding object using the grounding just explained; this is combined into an average description score, which is weighted by the gesture score to yield the final score of this mapping for this candidate scene.

Figure 6 shows a simple example (constructed using a simplified coordinate system for the gesture) of how each variant would process a description of a single object. Each variant is applied to the three candidate scenes.

4 Experiment

4.1 A Corpus of Scene Descriptions

To elicit natural language descriptions, we generated 25 pentomino scenes as described above and illustrated in Figure 2. We asked two student assistants (native German speakers; not authors of the paper) to write down verbal descriptions of the scenes, following a specific template (*here there is DESCR, and RELATION is...*). With these data, we do not need to run the full pipeline as described above but rather simulate the output of ASR (to focus on the core of the model for the purposes of this paper).

Example (3) shows a sample description, in which $|_{NS}$ indicates the start of a scene description and $|_{NObj}$ indicates the start of an object description. In total, we collected 50 scene descriptions.

(3) a. $|_{NS}$ Hier ist $|_{NObj}$ ein pinkes z-ähnliches Zeichen und schräg rechts unten davon ist $|_{NObj}$ ein zweites pinkes z-ähnliches Zeichen und schräg rechts unten davon ist $|_{NObj}$ ein blaues L

b. $|_{NS}$ here is $|_{NObj}$ a pink Z and diagonally to the bottom right of it is $|_{NObj}$ a second pink Z and diagonally bottom right of it is $|_{NObj}$ a blue L

We also simulate the outcome of the gesture recognition module, by taking the actual positions of the described objects as gesture positions and then adding (normally distributed) noise to simulate sensor uncertainty. The likelihood model for mapping scores is learned by producing a large number of noisy "gesture positions" based on real positions (by adding 2D gaussian noise, $\mu = 0, \sigma = 0.1$), scoring these, and then running a kernel density estimation to learn which deviation scores given the true positions are more likely.

The modules that are simulated in the evaluation are grayed out in Figure 4. Again, this is done to focus on testing the representation variants; swapping in the actual ASR and gesture modules, which we do have separately but not yet integrated, will hopefully result only in quantitatively but not qualitatively different performance.

4.2 Evaluation

To evaluate the performance of the model and the variants A-C, we created a set of test scenes for each description (hence, resulting in 50 test retrieval tasks), in three variants (for Experiments 1–3 below). Each test set includes the scene that was actually described plus as distractors five other scenes randomly selected from the set of 25 scenes (Experiment 3). For Experiment 1, the distractor scenes are modified so that all objects have the same position; i.e., in these cases, gesture information cannot help make a distinction and all load is on the verbal content. For Experiment 2, the object positions are kept, but all objects are replaced to be identical to those from the intended scene; i.e., here verbal content cannot distinguish between scenes. We created these different sets to be able to evaluate the relative contributions of each modality (Experiments 1 & 2) as well as the joint performance (Experiment 3).

We run the pipeline on the description to build the representation (or rather, three different representations, according to variants A-C) and to use this representation to retrieve the described scene from the set of six candidate scenes. We give results below in terms of accuracy (ratio of correct retrievals) as well as mean reciprocal rank (MRR), which is computed as follows (and ranges in our case from $1/6$ (worst) to 1 (ideal):

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank(i)} \quad (6)$$

4.3 Results

Table 2 shows the results of the experiments. Experiment 1 shows that when only language contributes to the retrieval task, the representation variants can already achieve good performance, with Variant A (verbatim representation / word classifiers) having a slight edge on Variant C (representation through clustering). Going just with the gesture information, by design, performs on chance level here (top row). Experiment 2 shows that all three variants perform robustly only with gesture information: In many sets, gesture information alone already identifies the correct scene (top row, "gesture"). Language can improve over this in cases where gesture-alone computes the wrong mapping of description IDs and object IDs. Experiment 3 finally shows application to the unchanged test set with randomly selected distractors. Here, verbal information can contribute even more, and variants A and C show a perfomance that is much better than variant B. (Variant B suffers from data sparsity: e.g., in the training data, the shape U was often described as "C", and rarely as "U", which is the preferred description in our test data, leading to the wrong shape property being predicted.) Interestingly, "compressing" the information into a small number (here: 26) of clusters does not seem to have hurt the performance.

5 Related Work

As noted by Roy and Reiter (2005), language is never used in isolation; the meanings of words are learned based on how they are used in contexts–for our purposes here, *visual* contexts–where visually-perceiveable scenes are described (albeit scenes that are later visually perceived). This approach to semantics is known as *grounding*; work has been done by, inter alia, Gorniak and Roy (2004), Gorniak and Roy (2005), Reckman et al. (2010) where word meanings such as colour, shape, and spatial terms were learned by resolving referring expressions. Symbolic approaches to semantic meaning (e.g., first-order logic) do not model such perceptual word meanings well (Harnad, 1990; Steels and Kaplan, 1999); here we follow Harnad (1990) and Larsson (2013) and try to reconcile grounded semantics and symbolic approaches. In this pa-

		Experiment 1		Experiment 2		Experiment 3	
		ACC	MRR	ACC	MRR	ACC	MRR
Gesture		0.1	0.37	0.65	0.75	0.67	0.78
Gesture+Speech	A	**0.82**	**0.90**	0.70	0.78	0.84	0.91
	B	0.68	0.81	0.68	0.76	0.68	0.81
	C	0.80	0.89	**0.76**	**0.82**	**0.84**	**0.92**

Table 2: Results of the Experiments. Exp. 1: objects in same spatial configuration in all scenes (per retrieval task); Exp. 2: objects potentially in different configurations in scenes, but same three objects in all scenes; Exp. 3: potentially different objects and different locations in all scenes.

per, we extended earlier work in this area (Larsson, 2013; Kennington et al., 2015) by learning and applying these mappings in a navigation task.

Navigation tasks provide a natural environment for the development and application of such a model of grounded semantics, which have been the subject of a fair amount of recent research: In Levit and Roy (2007), later extended in Kollar et al. (2010), the meaning of words related to map-navigation such as "toward" and "between" were learned from interaction data. Vogel and Jurafsky (2010) applied reinforcement learning to the task of learning the mapping between words in direction descriptions and routes. Also, Artzi and Zettlemoyer (2013) learned a semantic abstraction from the interaction map-task data in the form of a combinatory categorical grammar. Though interesting in their own right, these tasks made some important simplifying assumptions that we go beyond in this paper: first, gestural information is never used to convey scene descriptions; second, the scene that is being described (from a bird's-eye view; here, scenes are perceived from a first-person perspective) is visually-present at the time the descriptions are being made; third, that the grounded semantics of a select subset of words are being learned. In this paper, gestures are considered, a description is heard and *later* applied to scenes, and all the word groundings are learned from data.

The work presented in this paper is a natural next step that goes beyond map-task navigation and is pshycholinguistically motivated. Kintsch and van Dijk (1978) suggest that readers (listeners) first represent exact words of a description (i.e., surface form), then interpret information (i.e., a *gist* of the description) and integrate that with their world knowledge (e.g., the knowledge about what red things look like, if the word "red" was used in the description). Moreover, Brunyé and Taylor (2008) (as well as some work cited

there) note that readers construct cohesive mental models of what a text describes, integrating time, space, causality, intention, and person- and object-related information. That is, readers progress beyond the text itself to represent the described situation; detailed information from an instruction or description is distorted in memory (Moar and Bower, 1983). In this paper, we have shown in our evaluation that this is indeed the case; in Experiment 3, Variant C held the description in a more compact form than a (stemmed) surface form and produced better scores than Variant A, which did.

6 Conclusions

We have presented a first attempt at providing an end-to-end model of the task of understanding verbal/gestural scene descriptions, where this understanding can be tested by application of the understanding in a real-world (visual) discrimination task. We have explored different ways of representing content, where we went from not compressing the description at all (storing sequences of (stemmed) words, as they occurred), over using pre-specified property symbols to learning a set of "concepts" automatically. The approach overall performed well, with gesture information providing a large amount of information, with verbal content in all variants further improving over that.

In future work, we will test if the performance of the clustering approach can be improved by providing a larger amount of training data. We will also integrate the steps that were simulated here (speech and gesture recognition), and will integrate the processing pipeline into an interactive system that can potentially clarify the scene description it receives, while building the representation and before having to apply it.

References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions. *Transactions of the ACL*, 1:49–62.

Tad T Brunyé and Holly A Taylor. 2008. Working memory in developing and applying mental models from spatial descriptions. *Journal of Memory and Language*, 58(3):701–729.

Charles Fillmore. 1982. Frame semantics. *Linguistics in the morning calm*, pages 111–137.

Peter Gorniak and Deb Roy. 2004. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470.

Peter Gorniak and Deb Roy. 2005. Probabilistic Grounding of Situated Speech using Plan Recognition and Reference Resolution. In *In Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005)*, pages 138–143.

Stevan Harnad. 1990. The Symbol Grounding Problem. *Physica D*, 42:335–346.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer, Dordrecht.

Casey Kennington, Spyros Kousidis, and David Schlangen. 2013. Interpreting Situated Dialogue Utterances: an Update Model that Uses Speech, Gaze, and Gesture Information. In *SIGdial 2013*.

Casey Kennington, Livia Dia, and David Schlangen. 2015. A Discriminative Model for Perceptually-Grounded Incremental Reference Resolution. In *Proceedings of IWCS*. Association for Computational Linguistics.

Walter Kintsch and Teun a. van Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review*, 85(5):363–394.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2010. Toward understanding natural language directions. *Proceeding of the 5th ACMIEEE international conference on Humanrobot interaction HRI 10*, page 259.

Spyridon Kousidis, Thies Pfeiffer, Zofia Malisz, Petra Wagner, and David Schlangen. 2012. Evaluating a minimally invasive laboratory architecture for recording multimodal conversational data. In *Proceedings of the Interdisciplinary Workshop on Feedback Behaviors in Dialog, INTERSPEECH2012 Satellite Workshop*, pages 39–42.

Spyros Kousidis, Thies Pfeiffer, and David Schlangen. 2013. MINT . tools : Tools and Adaptors Supporting Acquisition , Annotation and Analysis of Multimodal Corpora. In *Proceedings of Interspeech 2013*, pages 2649–2653, Lyon, France. ISCA.

S Larsson. 2013. Formal semantics for perceptual classification. *Journal of Logic and Computation*.

Alex Lascarides and Matthew Stone. 2009. A formal semantic analysis of gesture. *Journal of Semantics*, 26(4):393–449.

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

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

Ian Moar and Gordon H Bower. 1983. Inconsistency in spatial knowledge. *Memory & Cognition*, 11(2):107–113.

Hilke Reckman, Jeff Orkin, and Deb Roy. 2010. Learning meanings of words and constructions, grounded in a virtual game. *Semantic Approaches in Natural Language Processing*, page 67.

Deb Roy and Ehud Reiter. 2005. Connecting language to the world. *Artificial Intelligence*, 167(1-2):1–12.

Laura F Schneider and Holly a. Taylor. 1999. How do you get there from here? Mental representations of route descriptions. *Applied Cognitive Psychology*, 13(September 1998):415–441.

Luc Steels and Frederic Kaplan. 1999. Situated grounded word semantics. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2, pages 862–867.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Adam Vogel and Dan Jurafsky. 2010. Learning to Follow Navigational Directions. In *Proceedings of ACL*, pages 806–814.