Semantic Parsing for Text to 3D Scene Generation

Angel X. Chang, Manolis Savva and **Christopher D. Manning**
Computer Science Department, Stanford University
angelx,msavva,manning@cs.stanford.edu

Figure 1: Generated scene for "There is a room with a chair and a computer." Note that the system infers the presence of a desk and that the computer should be supported by the desk.

1 Introduction

We propose text-to-scene generation as an application for semantic parsing. This is an application that grounds semantics in a virtual world that requires understanding of common, everyday language. In text to scene generation, the user provides a textual description and the system generates a 3D scene. For example, Figure 1 shows the generated scene for the input text "there is a room with a chair and a computer". This is a challenging, open-ended problem that prior work has only addressed in a limited way.

Most of the technical challenges in text to scene generation stem from the difficulty of mapping language to formal representations of visual scenes, as well as an overall absence of real world spatial knowledge from current NLP systems. These issues are partly due to the omission in natural language of many facts about the world. When people describe scenes in text, they typically specify only important, relevant information. Many common sense facts are unstated (e.g., chairs and desks are typically on the floor). Therefore, we focus on inferring implicit relations that are likely to hold even if they are not explicitly stated by the input text.

Text to scene generation offers a rich, interactive environment for grounded language that is familiar to everyone. The entities are common, everyday objects, and the knowledge necessary to address this problem is of general use across many domains. We present a system that leverages user interaction with 3D scenes to generate training data for semantic parsing approaches.

Previous semantic parsing work has dealt with grounding text to physical attributes and relations (Matuszek et al., 2012; Krishnamurthy and Kollar, 2013), generating text for referring to objects (FitzGerald et al., 2013) and with connecting language to spatial relationships (Golland et al., 2010; Artzi and Zettlemoyer, 2013). Semantic parsing methods can also be applied to many aspects of text to scene generation. Furthermore, work on parsing instructions to robots (Matuszek et al., 2013; Tellex et al., 2014) has analogues in the context of discourse about physical scenes.

In this extended abstract, we formalize the text to scene generation problem and describe it as a task for semantic parsing methods. To motivate this problem, we present a prototype system that incorporates simple spatial knowledge, and parses natural text to a semantic representation. By learning priors on spatial knowledge (e.g., typical positions of objects, and common spatial relations) our system addresses inference of implicit spatial constraints. The user can interactively manipulate the generated scene with textual commands, enabling us to refine and expand learned priors.

Our current system uses deterministic rules to map text to a scene representation but we plan to explore training a semantic parser from data. We can leverage our system to collect user interactions for training data. Crowdsourcing is a promising avenue for obtaining a large scale dataset.

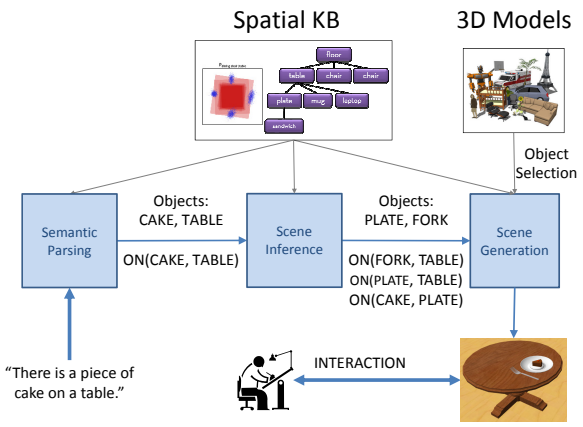

Figure 2: Illustration of our system architecture.

2 Task Definition

We define text to scene generation as the task of taking text describing a scene as input, and generating a plausible 3D scene described by that text as output. More concretely, we parse the input text into a *scene template*, which places constraints on what objects must be present and relationships between them. Next, using priors from a spatial knowledge base, the system expands the scene template by inferring additional implicit constraints. Based on the scene template, we select objects from a dataset of 3D models and arrange them to generate an output scene.

After a scene is generated, the user can interact with the scene using both textual commands and mouse interactions. During interaction, semantic parsing can be used to parse the input text into a sequence of *scene interaction commands*. See Figure 2 for an illustration of the system architecture. Throughout the process, we need to address grounding of language to: 1) actions to be performed, 2) objects to be instantiated or manipulated, and 3) constraints on the objects.

2.1 Scene Template

A scene template $\mathcal{T} = (\mathcal{O}, \mathcal{C})$ consists of a set of object descriptions $\mathcal{O} = \{o_1, \ldots, o_n\}$ and constraints $\mathcal{C} = \{c_1, \ldots, c_k\}$ on the relationships between the objects. For each object o_i, we identify properties associated with it such as category label, basic attributes such as color and material, and number of occurrences in the scene. Based on the object category and attributes, and other words in the noun phrase mentioning the object, we identify a set of associated keywords to be used later for querying the 3D model database. Spatial relations between objects are extracted as predicates of the form $on(o_i, o_j)$ or $left(o_i, o_j)$ where o_i and o_j are recognized objects.

As an example, given the input "*There is a room with a desk and a red chair. The chair is to the left of the desk.*" we extract the following objects and spatial relations:

Objects	category	attributes	keywords
o_0	room		room
o_1	desk		desk
o_2	chair	*color*:red	chair, red

Relations: $left(o_2, o_1)$

2.2 Scene Interaction Commands

During interaction, we parse textual input provided by the user into a sequence of commands with relevant parts of the scene as arguments. For example, given a scene $\mathcal{S}$, we use the input text to identify a subset of relevant objects matching $X = \{\mathcal{O}_s, \mathcal{C}_s\}$ where $\mathcal{O}_s$ is the set of object descriptions and $\mathcal{C}_s$ is the set of object constraints. Commands can then be resolved against this argument to manipulate the scene state: $Select(X)$, $Remove(X)$, $Insert(X)$, $Replace(X, Y)$, $Move(X, \Delta X)$, $Scale(X, \Delta X)$, and $Orient(X, \Delta X)$. X and Y are semantic representations of objects, while ΔX is a change to be applied to X, expressed as either a target condition ("put the lamp on the table") or a relative change ("move the lamp to the right").

These basic operations demonstrate possible scene manipulations through text. This set of operations can be enlarged to cover manipulation of parts of objects ("make the seat of the chair red"), and of the viewpoint ("zoom in on the chair").

2.3 Spatial Knowledge

One of the richest sources of spatial knowledge is 3D scene data. Prior work by (Fisher et al., 2012) collected 133 small indoor scenes created with 1723 3D Warehouse models. Based on their approach, we create a spatial knowledge base with priors on the static support hierarchy of objects in scenes[1], their relative positions and orientations. We also define a set of spatial relations such as *left, right, above, below, front, back, on top of, next to, near, inside, and outside*. Table 1 gives examples of the definitions of these spatial relations.

We use a 3D model dataset collected from Google 3D Warehouse by prior work in scene syn-

[1] A static support hierarchy represents which objects are likely to support which other objects on their surface (e.g., the floor supports tables, tables support plates).

Relation	$P(relation)$
inside(A,B)	$\frac{Vol(A \cap B)}{Vol(A)}$
right(A,B)	$\frac{Vol(A \cap \text{right}\ (B))}{Vol(A)}$
near(A,B)	$\mathbb{1}(dist(A,B) < t_{near})$

Table 1: Definitions of spatial relation using object bounding box computations.

thesis and containing about 12490 mostly indoor objects (Fisher et al., 2012). These models have text associated with them in the form of names and tags, and category labels. In addition, we assume the models have been scaled to physically plausible sizes and oriented with consistent up and front direction (Savva et al., 2014). All models are indexed in a database so they can be queried at runtime for retrieval.

3 System Description

We present how the parsed representations are used by our system to demonstrate the key issues that have to be addressed during text to scene generation. Our current implementation uses a simple deterministic approach to map text to the scene template and user actions on the scene. We use the Stanford CoreNLP pipeline[2] to process the input text and use rules to match dependency patterns.

3.1 Scene generation

During scene generation, we want to construct the most likely scene given the input text. We first parse the text into a scene template and use it to select appropriate models from the database. We then perform object layout and arrangement given the priors on spatial knowledge.

Scene Template Parsing We use the Stanford coreference system to determine when the same object is being referred to. To identify objects, we look for noun phrases and use the head word as the category, filtering with WordNet (Miller, 1995) to determine which objects are visualizable (under the physical object synset, excluding locations). To identify properties of the objects, we extract other adjectives and nouns in the noun phrase. We also match syntactic dependency patterns such as "X is made of Y" to extract more attributes and keywords. Finally, we use dependency patterns to extract spatial relations between objects.

[2] http://nlp.stanford.edu/software/corenlp.shtml

Figure 3: Select "a blue office chair" and "a wooden desk" from the models database

Object Selection Once we have the scene template, we use the keywords associated with each object to query the model database. We select randomly from the top 10 results for variety and to allow the user to regenerate the scene with different models. This step can be enhanced to take into account correlations between objects (e.g., a lamp on a table should not be a floor lamp model). See Figure 3 for an example of object selection.

Object Layout Given the selected models, the source scene template, and priors on spatial relations, we find an arrangement of the objects within the scene that maximizes the probability of the layout under the given scene template.

3.2 Scene Interaction

Here we address parsing of text after a scene has been generated and during interaction sessions.

Command Parsing We deterministically map verbs to possible actions as shown in Table 2. Multiple actions are possible for some verbs (e.g., "place" and "put" can refer to either $Move$ or $Insert$). To differentiate between these, we assume new objects are introduced with the indefinite article "a" whereas old ones are modified with the definite article "the".

Object Resolution To allow interaction with the scene, we must resolve references to objects within a scene. Objects are disambiguated by category and view-centric spatial relations. In addition to matching objects by their categories, we use the WordNet hierarchy to handle hyponym or hypernym referents. Depending on the current view, spatial relations such as "left" or "right" can refer to different objects (see Figure 4).

Scene Modification Based on the action we need to appropriately modify the current scene.

verb	Action	Example Text	Example Parse
generate	Generate	generate a room with a desk and a lamp	$Generate(\{room,desk,lamp\},\{\})$
select	Select	select the chair on the right of the table	$Select(\{lamp\},\{right(lamp,table)\})$
add, insert	Insert	add a lamp to the table	$Insert(\{lamp\},\{on(lamp,table)\})$
delete, remove	Remove	remove the lamp	$Remove(\{lamp\})$
move	Move	move the chair to the left	$Move(\{chair\},\{left(chair)\})$
place, put	Move, Insert	put the lamp on the table	$Move(\{lamp\},\{on(lamp,table)\})$
replace	Replace	replace the lamp with a vase	$Replace(\{lamp\},\{vase\})$

Table 2: Mapping of verbs to possible actions.

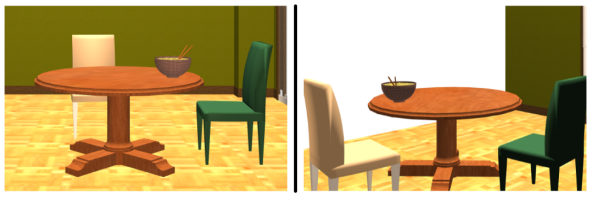

Figure 4: **Left**: chair is selected by "chair to the right of the table" or "object to the right of the table", but not selected by "cup to the right of the table". **Right**: Different view results in a different chair selection for "chair to the right of the table".

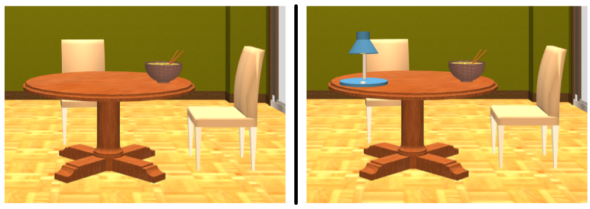

Figure 5: **Left**: initial scene. **Right**: after input "Put a lamp on the table".

We do this by maximizing the probability of a new scene template given the requested action and previous scene template (see Figure 5 for an example).

4 Future Directions

We described a system prototype to motivate approaching text to scene generation as a semantic parsing application. While this prototype illustrates inference of implicit constraints using prior knowledge, it still relies on hand coded rules for mapping text to the scene representation. This is similar to most previous work on text to scene generation (Winograd, 1972; Coyne and Sproat, 2001) and limits handling of natural language. More recently, (Zitnick et al., 2013) used data to learn how to ground sentences to a CRF representing 2D clipart scenes. Similarly, we plan to investigate using data to learn how to ground sentences to 3D scenes.

Spatial knowledge can be helpful for resolving ambiguities during parsing. For instance, from spatial priors of object positions and reasoning with physical constraints we can disambiguate the attachment of "next to" in "there is a book on the table next to the lamp". The book and lamp are likely on the table and thus $next\_to(book, lamp)$ should be more likely.

User interaction is a natural part of text to scene generation. We can leverage such interaction to obtain data for training a semantic parser. Every time the user issues a command, the user can indicate whether the result of the interaction was correct or not, and optionally provide a rating. By keeping track of these scene interactions and the user ratings we can construct a corpus of tuples containing: user action, parsed scene interaction, scene operation, scene state before and after the operation, and rating by the user. By building up such a corpus over multiple interactions and users, we obtain data for training semantic parsers.

References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*.

Bob Coyne and Richard Sproat. 2001. WordsEye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. 2012. Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics*.

Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. 2013. Learning distributions over logical forms for referring expression generation. In *Proceedings of the Conference on EMNLP*.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on EMNLP*.

Jayant Krishnamurthy and Thomas Kollar. 2013. Jointly learning to parse and perceive: Connecting

natural language to the physical world. *Transactions of the Association for Computational Linguistics*.

Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *International Conference on Machine Learning*.

Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. 2013. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*.

G.A. Miller. 1995. WordNet: a lexical database for english. *CACM*.

Manolis Savva, Angel X. Chang, Gilbert Bernstein, Christopher D. Manning, and Pat Hanrahan. 2014. On being the right scale: Sizing large collections of 3D models. *Stanford University Technical Report CSTR 2014-03*.

Stefanie Tellex, Pratiksha Thaker, Joshua Joseph, and Nicholas Roy. 2014. Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*.

Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*.

C. Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. 2013. Learning the visual interpretation of sentences. In *IEEE Intenational Conference on Computer Vision (ICCV)*.