3D Scene Retrieval from Text with Semantic Parsing

Will Monroe[*]
Stanford University
Computer Science Department
Stanford, CA 94305, USA
wmonroe4@cs.stanford.edu

Abstract

We examine the problem of providing a natural-language interface for retrieving 3D scenes from descriptions. We frame the natural language understanding component of this task as a semantic parsing problem, wherein we first generate a structured meaning representation of a description and then use the meaning representation to specify the space of acceptable scenes. Our model outperforms a one-vs.-all logistic regression classifier on synthetic data, and remains competitive on a large, real-world dataset. Furthermore, our model learns to favor meaning representations that take into account more of the natural language meaning and structure over those that ignore words or relations between them.

1 Introduction

We look at the task of *3D scene retrieval*: given a natural-language description and a set of 3D scenes, identify a scene matching the description. Geometric specifications of 3D scenes are part of the craft of many graphical computing applications, including computer animation, games, and simulators. Large databases of such scenes have become available in recent years as a result of improvements in the ease of use of tools for 3D scene design. A system that can identify a 3D scene from a natural language description is useful for making such databases of scenes readily accessible. Natural language has evolved to be well-suited to describing our (three-dimensional) world, and it provides a convenient way of specifying the space of acceptable scenes: a description of a physical environment encodes logical propositions about the space of environments a speaker is describing.

"brown room with a refrigerator in the back corner"

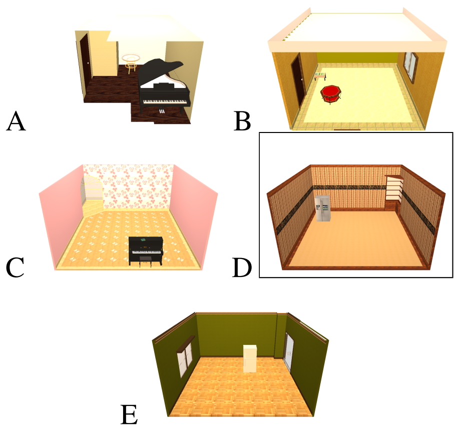

Figure 1: One instance of the scene retrieval task.

This logical structure of scene descriptions suggests that *semantic parsing* is an appropriate framework in which to understand the problem of retrieving scenes based on their natural-language descriptions. Semantic parsing (Zelle and Mooney, 1996; Tang and Mooney, 2001; Zettlemoyer and Collins, 2005) is the problem of extracting logical forms from natural language.

[*] This project is part of a joint research project with Angel Chang (Computer Science) and Christopher Potts (Linguistics). I have omitted their names from the author section to avoid confusion, as neither is a CS 229 student; however, their contributions are substantial. Angel implemented the infrastructure for working with 3D scenes and collecting data via Mechanical Turk. Prof. Potts has been our primary faculty advisor for this research project.

$$\begin{aligned}
&S \to P\,F\,P\\
&F \to O\,(P\,O)^*\\
&P \to \Sigma^*\\
&O \to \text{room}|\text{green}|\text{weird}|\ldots
\end{aligned}$$

Figure 2: Context-free grammar for parsing scene descriptions.

A large and growing body of literature from the computer vision and information retrieval communities exists surrounding the task of scene retrieval over images and video given descriptions (Sebe et al., 2003; Datta et al., 2008; Snoek and Worring, 2008). However, relatively little work has looked at identifying 3D scenes from natural language descriptions. 3D scene representations offer advantages in the form of a highly structured, segmented input that is semantically annotated.

Other work has looked at *generation* of 3D scenes from text. WordsEye (Coyne and Sproat, 2001) pioneered the approach of visualizing natural language by constructing a 3D scene that represents its meaning. More recently, Chang et al. (2014) built a system for text-to-scene generation which can learn scene synthesis from a database of human-built scenes (Fisher et al., 2012) and be trained interactively, updating the weights for object support hierarchy priors in response to user corrections. This helps ensure that generated scenes are realistic, by sampling from an estimate of the density of human-generated scenes that is trained on both positive and negative examples.

Both of these systems notably lack sophisticated approaches for extracting meaning from natural language descriptions. Our work relates to this goal in that we define a space of meaning representations that can be used for both 3D scene retrieval and 3D scene generation. We aims to fill the natural language understanding gap in this work by applying a state-of-the-art semantic parser to map textual input to these meaning representations.

2 Models

We define the *scene retrieval* problem as follows: given $x^{(i)}$, a natural-language description of a 3D scene, identify the scene $y^{(i)}$ that inspired the description from among a set of scenes $Y^{(i)}$ consisting of $y^{(i)}$ and one or more distractor scenes.

2.1 Logistic regression

We train and evaluate two models for solving this problem. The first is a one-vs.-all logistic regression (LR) classifier, which learns to estimate the probability of a scene being correct:

$$h_\theta(y|x) = \frac{1}{1 + \exp[-\phi(x,y)^T\theta]}$$

In training the LR model, we maximize the L$_2$-regularized log likelihood of the dataset, counting each true scene and each distractor scene as one example:

$$\mathcal{O}(\mathbf{x},\mathbf{y}) = \frac{\lambda}{2}||\theta||_2^2 + \sum_{i=1}^{m}\sum_{y\in Y^{(i)}}\Big(\mathbb{1}\left\{y = y^{(i)}\right\}\log h_\theta(y|x^{(i)}) + \mathbb{1}\left\{y \neq y^{(i)}\right\}\log[1 - h_\theta(y|x^{(i)})]\Big)$$

At test time, the model predicts the scene with the highest estimated probability:

$$\hat{y}^{(i)} = \operatorname*{arg\,max}_{y\in Y^{(i)}} h_\theta(y|x^{(i)})$$

2.2 Semantic parsing

The second model we consider is a structured prediction model that uses the SEMPRE semantic parsing framework (Berant et al., 2013) to derive a structured logical representation (z, a latent variable) of the meaning of a scene description. SEMPRE uses a structured softmax regression model with features defined over utterance–logical form pairs:

$$h_\theta(z|x) = \frac{\exp[\phi(x,z)^T\theta]}{\sum_{z'\in D(x;\theta)}\exp[\phi(x,z')^T\theta]}$$

The space of logical forms compatible with an input utterance $D(x)$ is in general exponentially large, so SEMPRE approximates this sum using a beam search algorithm (guided by the learned parameters θ) to search a subset $\tilde{D}(x;\theta)$ of possible outputs.

For each logical form z, our model uses deterministic scoring rules to select the scene that best matches the constraints expressed in the logical form:

$$[\![z]\!] = \arg\max_{y \in Y^{(i)}} \mathrm{Score}(y, z)$$

The training objective is to maximize the L$_1$-regularized log likelihood of the correct scenes, marginalized over z:

$$\mathcal{O}(\mathbf{x}, \mathbf{y}) = \lambda ||\theta||_1 + \sum_{i=1}^{m} \log \sum_{z \in \tilde{D}(x^{(i)};\theta)} \mathbb{1}\left\{ [\![z]\!] = y^{(i)} \right\} h_\theta(z|x^{(i)})$$

The final prediction of the model is the scene selected by the logical form on the beam that is the most likely according to the model:

$$\hat{y}^{(i)} = \left[\!\!\left[\arg\max_{z \in D(x^{(i)};\theta)} h_\theta(z|x^{(i)}) \right]\!\!\right]$$

$D(x)$, the true set of logical forms compatible with a scene description x, is defined by a context-free grammar over the input augmented with semantic interpretations of each grammar rule. Figure 2 shows the simple grammar we use. Any terminal and nonterminal symbol in this grammar can be surrounded by zero or more tokens that are skipped (not used in the output logical form). The symbol O represents the lexicon for this grammar, which consists of phrases mapped to subsets of a database of 3D models, either by category (e.g. “chair” is mapped to the set of objects categorized by humans as “chairs” in the database) or by model ID (e.g. “crazy poster” is mapped to the specific model `poster463`). Our logical forms then consist of a conjunction of predicates denoting the existence of at least one object from each subset. For example, one possible logical form might look like:

```
(and category:room
     model:refrigerator192)
```

2.2.1 Features

Both models use binary-valued features indicating the co-occurrence of a unigram or bigram and an object category or model ID. The logistic regression model uses features defined over every unigram and bigram in the input, a total of 15,000 features on the development dataset and 236,000 features on the full dataset. SEMPRE uses unigram and bigram features only for entries in the lexicon O (approximately 900 for the development dataset and 2,000 for the full dataset). These features are very sparse; for a given input x containing n words, at most $2n - 1$ of these features are non-zero in the LR model, and in the SEMPRE model, at most $(2n - 1)|z|$ are non-zero, where $|z|$ is the number of object descriptions in a given logical form.

For the SEMPRE model, we also have features over the logical forms $|z|$. These consist of counts of words skipped by part of speech and counts of grammar rules applied in constructing the logical form. These features allow the algorithm to adjust the fraction of the natural language input that is used or ignored in building the output as well as the complexity of the output logical forms. On the development set, the model stored weights for 19 rule application features and 20 POS-skipping features; for the full dataset, the numbers were 26 and 34, respectively.

3 Experiments

3.1 Datasets

We collected a dataset of 860 3D scenes and 1,735 descriptions of these scenes from workers on Mechanical Turk. One set of Turkers are shown a sentence (one of 60 seed sentences we wrote describing simple configurations of interior scenes) and asked to build a scene using the WebSceneStudio interface (the program used by Fisher et al. (2012) to build their scene synthesis database). Other Turkers are then asked to describe each of these scenes in one to two sentences.

Our development was done on a smaller dataset of 302 machine-generated scenes, using an existing scene generation system (Chang et al., 2014). These are created from a set of 36 logical forms, some of which were augmented with spatial constraints, such as `(and category:candle category:table)` and `(on_top_of category:candle category:table)`. For each of these, we generated a number of scenes that satisfy the requirements expressed by the logical form (e.g. that have both a candle and a table), manually filtering scenes to eliminate low-quality models and unphysical object placement. We then wrote one description for each of these 302 scenes.

Model	Train	Test
Random	20%	20%
LR-raw-uni	100%	55%
LR-raw	100%	56%
LR-lem	100%	57%
SEMPRE-raw	40.6%	39%
SEMPRE-lem	46.0%	47%
SEMPRE-lex-raw	81.2%	57%
SEMPRE-lex-lem	86.6%	**66%**
Size	202	100

Table 1: Development dataset results. **raw** and **lem** indicate using unprocessed and lemmatized text for unigram and bigram features; **uni** indicates unigram features only; **lex** indicates use of a lexicon assembled from the training set instead of written by hand.

Model	Train	Test
Random	20%	20%
LR	99.9%	**52.3%**
SEMPRE	70.7%	41.3%
Size	1,437	298

Table 2: Results on the Mechanical Turk dataset.

3.2 Results

Table 1 shows the performance of our model on the development dataset. Our best semantic parsing model achieves 66% accuracy at distinguishing a scene that matches a description from four distractors (chance is 20%) on a held-out set of 100 scenes from this development dataset. The one-vs.-all classifier achieves 57% accuracy on the held-out dev set.

The fact that the LR model is able to perfectly classify the development training set indicated that it was overfitting. To reduce variance, we looked at the effects of regularization and training set size on model performance.

Figure 3 shows held-out dev set accuracy as a function of the regularization constant λ. We found that including regularization only barely improved the LR model, and hurt performance of the SEMPRE model across the board. (Note that LR uses L_2 regularization and SEMPRE uses L_1 regularization, so the constants are not directly comparable, but in both cases larger numbers represent a stronger prior.)

Figure 4 shows the learning curves of the two models on the development set. The slopes of the

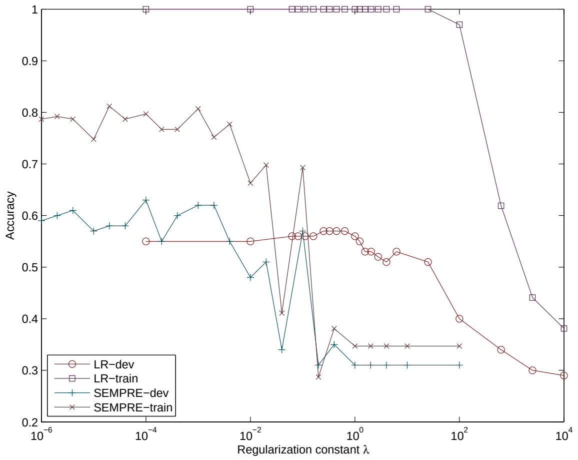

Figure 3: Effects of regularization on development set accuracy. We found that regularization did not improve performance on either model.

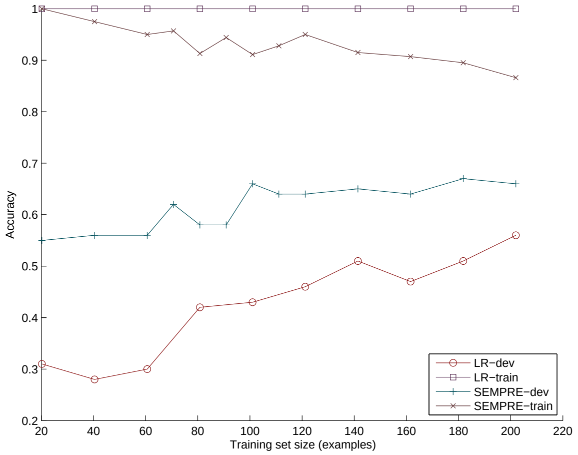

Figure 4: Learning curve on the development dataset. The upward slope indicated that additional data was likely to improve performance; the steeper line for the LR model suggested (correctly) that LR would outperform SEMPRE on the larger dataset.

learning curves suggest that the SEMPRE model does well in the low-data setting, but the LR model benefits from the larger dataset. This can be explained by the fact that SEMPRE has a fixed lexicon that encodes some amount of prior knowledge about word meanings; as the training data grows, this lexicon becomes less capable of accounting for the variety of language found in the data, whereas the LR model continues to add features and learn from the additional data.

The results on our dataset collected from Mechanical Turk (Figure 2) support this conclusion: the LR model performed better than the semantic parsing model on this larger dataset, achieving 52.3% accuracy on the test set compared to 41.3% for the semantic parsing model (chance is again 20%).

4 Discussion

On the smaller dev set, the semantic parsing model had higher performance than the simple classifier. However, with a larger dataset, logistic regression outperforms the structured model. This discrepancy is best explained by a combination of two factors: firstly, additional data provides a greater benefit to the LR model than to the semantic parsing model, as mentioned in Section 3.2. Secondly, the larger dataset contains descriptions collected from many workers, whereas the descriptions in the smaller dataset were written only by two people. This naturally leads to greater linguistic diversity in the full dataset, making the problem harder for both models (as we can see from the lower test set scores in Figure 2) but disproportionately affecting the semantic parsing model because of its limited capacity for encoding linguistic variation in the logical forms.

One promising outcome we observed is that the semantic parsing model gives negative weights to the word skipping features (with the exception of verbs, wh-pronouns, and punctuation) and the grammar rule application features with counts of zero. This indicates that paying attention to all of the language in a description and producing outputs with complex structure are empirically beneficial.

5 Future Work

For future work, we plan to examine ways to speed up training times, so we can use a larger lexicon. We also plan to add new features and new kinds of meaning representations, incorporating spatial relationships between objects and other information about 3D models that is available in our model database such as human-written descriptions of models. Other attributes that can be extracted from the models that could be useful in features include color, size, and proportions.

The meaning representations produced by the SEMPRE model can also be used for *scene synthesis*, a line of work we plan to pursue over the coming months.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *EMNLP*.

Angel X. Chang, Manolis Savva, and Christopher D. Manning. 2014. Interactive learning of spatial knowledge for text to 3D scene generation. In *ACL-ILLVI*.

Bob Coyne and Richard Sproat. 2001. WordsEye: an automatic text-to-scene conversion system. In *SIGGRAPH*.

Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2):5.

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. 2012. Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):135.

Nicu Sebe, Michael S. Lew, Xiang Zhou, Thomas S. Huang, and Erwin M. Bakker. 2003. The state of the art in image and video retrieval. In *Image and Video Retrieval*, pages 1–8. Springer.

Cees GM Snoek and Marcel Worring. 2008. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322.

Lappoon R. Tang and Raymond J. Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Machine Learning: ECML 2001*. Springer.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI*, pages 1050–1055.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*.