Generating Semantically Precise Scene Graphs from Textual Descriptions for Improved Image Retrieval

Sebastian Schuster, Ranjay Krishna, Angel Chang,
Li Fei-Fei, and **Christopher D. Manning**

Stanford University, Stanford, CA 94305
{sebschu,rak248,angelx,feifeili,manning}@stanford.edu

Abstract

Semantically complex queries which include attributes of objects and relations between objects still pose a major challenge to image retrieval systems. Recent work in computer vision has shown that a graph-based semantic representation called a *scene graph* is an effective representation for very detailed image descriptions and for complex queries for retrieval. In this paper, we show that scene graphs can be effectively created automatically from a natural language scene description. We present a rule-based and a classifier-based scene graph parser whose output can be used for image retrieval. We show that including relations and attributes in the query graph outperforms a model that only considers objects and that using the output of our parsers is almost as effective as using human-constructed scene graphs (Recall@10 of 27.1% vs. 33.4%). Additionally, we demonstrate the general usefulness of parsing to scene graphs by showing that the output can also be used to generate 3D scenes.

1 Introduction

One of the big remaining challenges in image retrieval is to be able to search for very specific images. The continuously growing number of images that are available on the web gives users access to almost any picture they can imagine, but in order to find these images users have to be able to express what they are looking for in a detailed and efficient way. For example, if a user wants to find an image of *a boy wearing a t-shirt with a plane on it*, an image retrieval system has to understand that the image should contain a boy who is wearing a shirt and that on that shirt is a picture of a plane.

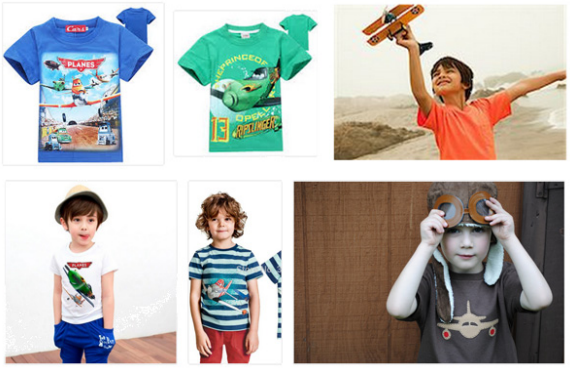

Figure 1: Actual results using a popular image search engine (top row) and ideal results (bottom row) for the query *a boy wearing a t-shirt with a plane on it.*

Keyword-based image retrieval systems are clearly unable to deal with the rich semantics of such a query (Liu et al., 2007). They might be able to retrieve images that contain a boy, a t-shirt and a plane but they are unable to interpret the relationships and attributes of these objects which is crucial for retrieving the correct images. As shown in Figure 1, a possible but incorrect combination of these objects is that a boy is wearing a t-shirt and playing with a toy plane.

One proposed solution to these issues is the mapping of image descriptions to multi-modal embeddings of sentences and images and using these embeddings to retrieve images (Plummer et al., 2015; Karpathy et al., 2014; Kiros et al., 2015; Mao et al., 2015; Chrupala et al., 2015). However, one problem of these models is that they are trained on single-sentence captions which are typically unable to capture the rich content of visual scenes in their entirety. Further, the coverage of the description highly depends on the subjectivity of human perception (Rui et al., 1999). Certain details such as whether there is a plane on the boy's shirt or not might seem irrelevant to the per-

Proceedings of the 2015 Workshop on Vision and Language (VL'15), pages 70–80,
Lisbon, Portugal, 18 September 2015.

son who writes the caption, but for another user this difference might determine whether a result is useful or not.

Johnson et al. (2015) try to solve these problems by annotating images with a graph-based semantic representation called a *scene graph* which explicitly captures the objects in an image, their attributes and the relations between objects. They plausibly argue that paragraph-long image descriptions written in natural language are currently too complex to be mapped automatically to images and instead they show that very detailed image descriptions in the form of scene graphs can be obtained via crowdsourcing. They also show that they can perform semantic image retrieval on unannotated images using partial scene graphs.

However, one big shortcoming of their model is that it requires the user to enter a query in the form of a scene graph instead of an image description in natural language which is unlikely to find widespread adoption among potential users. To address this problem, we propose a new task of parsing image descriptions to scene graphs which can then be used as a query for image retrieval.

While our main goal is to show the effectiveness of parsing image descriptions for image retrieval, we believe that scene graphs can be a useful intermediate representation for many applications that involve text and images. One great advantage of such an intermediate representation is the resulting modularity which allows independent development, improvement and reuse of NLP, vision and graphics subsystems. For example, we can reuse a scene graph parser for systems that generate 2D-scenes (Zitnick et al., 2013) or 3D-scenes (Chang et al., 2014) which require input in the form of similar graph-based representations to which a scene graph can be easily converted.

In this paper, we introduce the task of parsing image descriptions to scene graphs. We build and evaluate a rule-based and a classifier-based scene graph parser which map from dependency syntax representations to scene graphs. We use these parsers in a pipeline which first parses an image description to a scene graph and then uses this scene graph as input to a retrieval system. We show that such a pipeline outperforms a system which only considers objects in the description and we show that the output of both of our parsers is almost as effective as human-constructed scene graphs in retrieving images. Lastly, we demonstrate the more general applicability of our parsers by generating 3D scenes from their output.

We make our parsers and models available at http://nlp.stanford.edu/software/scenegraph-parser.shtml.

2 Task Description

Our overall task is retrieving images from image descriptions which we split into two sub-tasks: Parsing the description to scene graphs and retrieving images with scene graphs. In this paper, we focus exclusively on the first task. For the latter, we use a reimplementation of the system by Johnson et al. (2015) which we briefly describe in the next section.

2.1 Image Retrieval System

The image retrieval system by Johnson et al. (2015) is based on a conditional random field (CRF) (Lafferty et al., 2001) model which – unlike the typical CRFs in NLP – is not a chain model but instead capturing image region proximity. This model ranks images based on how likely it is that a given scene graph is grounded to them. The model first identifies potential object regions in the image and then computes the most likely assignment of objects to regions considering the classes of the objects, their attributes and their relations. The likelihood of a scene graph being grounded to an image is then approximated as the likelihood of the most likely assignment of objects to regions.

2.2 Parsing to Scene Graphs

The task of parsing image descriptions to scene graphs is defined as following. Given a set of object classes C, a set of relation types R, a set of attribute types A, and a sentence S we want to parse S to a scene graph $G = (O, E)$. $O = \{o_1, ..., o_n\}$ is a set of objects mentioned in S and each o_i is a pair (c_i, A_i) where $c_i \in C$ is the class of o_i and $A_i \subseteq A$ are the attributes of o_i. $E \subseteq O \times R \times O$ is the set of relations between two objects in the graph. For example, given the sentence $S =$ *"A man is looking at his black watch"* we want to extract the two objects $o_1 = (\textit{man}, \emptyset)$ and $o_2 = (\textit{watch}, \{\textit{black}\})$, and the relations $e_1 = (o_1, \textit{look at}, o_2)$ and $e_2 = (o_1, \textit{have}, o_2)$. The sets C, R and A consist of all the classes and types which are present in the training data.

2.3 Data

We reuse a dataset which we collected for a different task using Amazon Mechanical Turk (AMT) in a similar manner as Johnson et al. (2015) and Plummer et al. (2015). We originally annotated 4,999 images from the intersection of the YFCC100m (Thomee et al., 2015) and Microsoft COCO (Lin et al., 2014b) datasets. However, unlike previous work, we split the process into two separate passes with the goal of increasing the number of objects and relations per image.

In the first pass, AMT workers were shown an image and asked to write a one sentence description of the entire image or any part of it. To get diverse descriptions, workers were shown the previous descriptions written by other workers for the same image and were asked to describe something about the image which had not been described by anyone else. We ensured diversity in sentence descriptions by a real-time BLEU score (Papineni et al., 2002) threshold between a new sentence and all the previous ones.

In the second pass, workers were presented again with an image and with one of its sentences. They were asked to draw bounding boxes around all the objects in the image which were mentioned in the sentence and to describe their attributes and the relations between them. This step was repeated for each sentence of an image and finally the partial scene graphs are combined to one large scene graph for each image. While the main purpose of the two-pass data collection was to increase the number of objects and relations per image, it also provides as a byproduct a mapping between sentences and partial scene graphs which gives us a corpus of sentence-scene graph pairs that we can use to train a parser.

2.3.1 Preprocessing

The AMT workers were allowed to use any label for objects, relations and attributes and consequently there is a lot of variation in the data. We perform several preprocessing steps to canonicalize the data. First, we remove leading and trailing articles from all labels. Then we replace all the words in the labels with their lemmata and finally we split all attributes with a conjunction such as *red and green* into two individual attributes.

We also follow Johnson et al. (2015) and discard all objects, relations and attributes whose class or type appears less than 30 times in the entire dataset for the following two reasons. First and foremost, computer vision systems require multiple training examples for each class and type to be able to learn useful generalizations, and second, rare classes and types are often a result of AMT workers making mistakes or not understanding the task properly. As we make the assumption that the scene graph of one sentence is complete, i.e., that it captures all the information of the sentence, we have to apply a more aggressive filtering which discards the entire scene graph of a sentence in case one of its objects, attributes or relations is discarded due to the threshold. In case we discard all sentences of an image, we discard the entire image from our data. Despite the aggressive filtering, the average number of objects, relations and attributes per image only drops by 30-45% and we only discard around 9% of the images (see Table 1).

	Raw	**Processed**	**Filtered**
Images	4,999	4,999	4,524
Sentences	88,188	88,188	50,448
Sentences per image	17.6	17.6	11.2
Object classes	18,515	15,734	798
Attribute types	7,348	6,442	277
Relation types	9,274	7,507	131
Objects per image	21.2	21.2	14.6
Attributes per image	16.2	16.4	10.7
Relations per image	18.6	18.6	10.3
Attributes per sent.	0.92	0.93	0.93
Relations per sent.	1.06	1.06	0.96

Table 1: Aggregate statistics of the raw, canonicalized (processed) and filtered datasets.

3 Scene Graph Parsers

We implement two parsers: a rule-based parser and a classifier-based parser. Both of our parsers operate on a linguistic representation which we refer to as a *semantic graph*. We obtain semantic graphs by parsing the image descriptions to dependency trees followed by several tree transformations. In this section, we first describe these tree transformations and then explain how our two parsers translate the semantic graph to a scene graph.

3.1 Semantic Graphs

A Universal Dependencies (de Marneffe et al., 2014) parse is in many ways close to a shallow semantic representation and therefore a good starting point for parsing image descriptions to scene

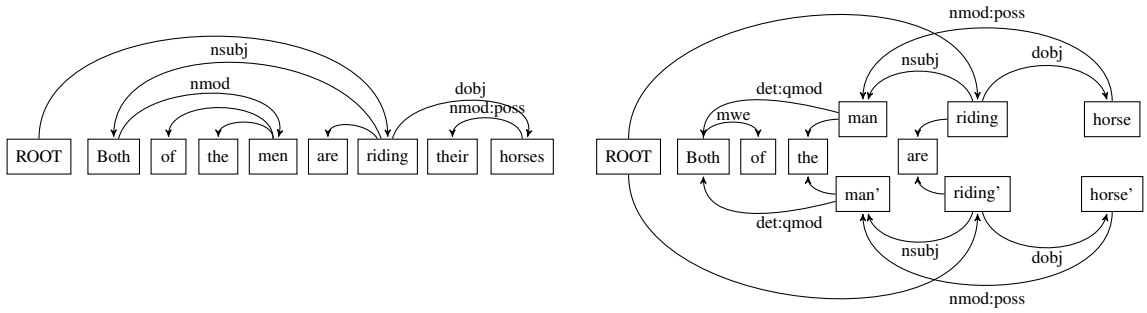

Figure 2: Dependency tree and final semantic graph of a sentence. *men* is promoted to be the subject; *men*, *riding*, and *horses* are duplicated; and *their* is deleted following coreference resolution.

graphs. Basic dependency trees, however, tend to follow the linguistic structure of sentences too closely which requires some post-processing of the parses to make them more useful for a semantic task. We start with the *enhanced* dependency representation output by the Stanford Parser v3.5.2 (Klein and Manning, 2003)[1] and then perform three additional processing steps to deal with complex quantificational modifiers, to resolve pronouns and to handle plural nouns.

3.1.1 Quantificational modifiers

Several common expressions with light nouns such as *a lot of* or *a dozen of* semantically act like quantificational determiners (Simone and Masini, 2014). From a syntactic point of view, however, these expressions are the head of the following noun phrase. While one of the principles of the Universal Dependencies representation is the primacy of content words (de Marneffe et al., 2014), light nouns are treated like any other noun. To make our dependency trees better suited for semantic tasks, we change the structure of all light noun expressions from a manually compiled list. We make the first word the head of all the other words in the expression and then make this new multi-word expression a dependent of the following noun phrase. This step guarantees that the semantic graph for *both cars* and for *both of the cars* have similar structures in which the semantically salient word *cars* is the head.

3.1.2 Pronoun resolution

Some image descriptions such as *"a bed with a pillow on it"* contain personal pronouns. To recover all the relations between objects in this sentence it is crucial to know that *it* refers to the object *a bed* and therefore we try to resolve all pronouns. We found in practice that document-level coreference systems (e.g. Lee et al. (2013)) were too conservative in resolving pronouns and hence we implement an intrasential pronoun resolver inspired by the first three rules of the Hobbs algorithm (Hobbs, 1978) which we modified to operate on dependency trees instead of constituency trees. We evaluate this method using 200 randomly selected image descriptions containing pronouns. Our pronoun resolver has an accuracy of 88.5% which is significantly higher than the accuracy of 52.8% achieved by the coreference system of Lee et al. (2013).

3.1.3 Plural nouns

Plural nouns are known to be a major challenge in semantics in general (Nouwen, 2015), and also in our task. One particular theoretical issue is the collective-distributive ambiguity of sentences with multiple plural nouns. For example, to obtain the intended distributive reading of *"three men are wearing jeans"* we have to extract three *man* objects and three *jeans* objects and we have to connect each *man* object to a different *jeans* object. On the other hand, to get the correct parse of *"three men are carrying a piano"* we probably want to consider the collective reading and extract only one *piano* object. A perfect model thus requires a lot of world knowledge. In practice, however, the distributive reading seems to be far more common so we only consider this case.

To make the dependency graph more similar to scene graphs, we copy individual nodes of the graph according to the value of their numeric modifier. We limit the number of copies per node to 20

[1]We augment the parser's training data with the Brown corpus (Marcus et al., 1993) to improve its performance on image descriptions which are often very different from sentences found in newswire corpora.

as our data only contains scene graphs with less than 20 objects of the same class. In case a plural noun lacks such a modifier we make exactly one copy of the node.

Figure 2 shows the original dependency tree and the final semantic graph for the sentence "*Both of the men are riding their horses*".

3.2 Rule-Based Parser

Our rule-based parser extracts objects, relations and attributes directly from the semantic graph. We define in total nine dependency patterns using Semgrex[2] expressions. These patterns capture the following constructions and phenomena:

- Adjectival modifiers
- Subject-predicate-object constructions and subject-predicate constructions without an object
- Copular constructions
- Prepositional phrases
- Possessive constructions
- Passive constructions
- Clausal modifiers of nouns

With the exception of possessives for which we manually add a *have* relation, all objects, relations and attributes are words from the semantic graph. For example, for the semantic graph in Figure 2, the *subject-predicate-object* pattern matches $man \xleftarrow{nsubj} riding \xrightarrow{dobj} horse$ and $man' \xleftarrow{nsubj} riding' \xrightarrow{dobj} horse'$. From these matches we extract two *man* and two *horse* objects and add *ride* relations to the two *man-horse* pairs. Further, the *possesive* pattern matches $man \xleftarrow{nmod:poss} horse$ and $man' \xleftarrow{nmod:poss} horse'$ and we add *have* relations to the two *man-horse* pairs.

3.3 Classifier-Based Parser

Our classifier-based parser consists of two components. First, we extract all candidate objects and attributes, and second we predict relations between objects and the attributes of all objects.

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

3.3.1 Object and Attribute Extraction

We use the semantic graph to extract all object and attribute candidates. In a first step we extract all nouns, all adjectives and all intransitive verbs from the semantic graph. As this does not guarantee that the extracted objects and attributes belong to known object classes or attribute types and as our image retrieval model can only make use of known classes and types, we predict for each noun the most likely object class and for each adjective and intransitive verb the most likely attribute type. To predict classes and types, we use an L_2-regularized maximum entropy classifier which uses the original word, the lemma and the 100-dimensional GloVe word vector (Pennington et al., 2014) as features.

3.3.2 Relation Prediction

The last step of the parsing pipeline is to determine the attributes of each object and the relations between objects. We consider both of these tasks as a pairwise classification task. For each pair (x_1, x_2) where x_1 is an object and x_2 is an object or an attribute we predict the relation y which can be any relation seen in the training data, or one of the two special relations *IS* and *NONE* which indicate that x_2 is an attribute of x_1 or no relation exists, respectively. We noticed that for most pairs for which a relation exists, x_1 and x_2 are in the same constituent, i.e. their lowest common ancestor is either one of the two objects or a word in between them. We therefore consider only pairs which satisfy this constraint to improve precision and to limit the number of predictions.

For the predictions, we use again an L_2-regularized maximum entropy classifier with the following features:

Object features The original word and lemma, and the predicted class or type of x_1 and x_2.

Lexicalized features The word and lemma of each token between x_1 and x_2. If x_1 or x_2 appear more than once in the sentence because they replace a pronoun, we only consider the words in between the closest mentions of x_1 and x_2.

Syntactic features The concatenated labels (i.e., syntactic relation names) of the edges in the shortest path from x_1 to x_2 in the semantic graph.

We only include objects in the scene graph which have at least one attribute or which are involved in at least one relation. The idea behind

that is to prevent very abstract nouns such as *setting* or *right* to be part of the scene graph which are typically not part of relations. However, we observed for around 30% of the sentences in the development set that the parser did not extract any relations or attributes from a sentence which resulted in an empty scene graph. In these cases, we include all candidate objects in the scene graph.

3.3.3 Training

As the scene graph's objects and attributes are not aligned to the sentence, we have to align them in an unsupervised manner. For each sentence, we extract object and attribute candidates from the semantic graph. For each object-relation-object triple or object-attribute pair in the scene graph we try to align all objects and attributes to a candidate by first checking for exact string match of the word or the lemma, then by looking for candidates within an edit distance of two, and finally by mapping the object or attribute and all the candidates to 100-dimensional GloVe word vectors and picking the candidate with the smallest euclidean distance. To limit the number of false alignments caused by annotators including objects in the scene graph that are not present in the corresponding sentence, we also compute the euclidean distances to all the other words in the sentence and if the closest match is not in the candidate set we discard the training example.

We use this data to train both of our classifiers. For the object and attribute classifier, we only consider the alignments between words in the description and objects or attributes in the graph.

For the relation predictor, we consider the complete object-relation-object and object-*is*-attribute triples. All the aligned triples constitute our positive training examples for a sentence. For all the object-object and object-attribute pairs without a relation in a sentence, we generate negative examples by assigning them a special $NONE$ relation. We sample from the set of $NONE$ triples to have the same number of positive and negative training examples.

4 Experiments

For our experiments, we split the data into training, development and held-out test sets of size 3,614, 454, and 456 images, respectively. Table 2 shows the aggregated statistics of our training and test sets. We compare our two parsers against the following two baselines.

	Train	**Dev**	**Test**
Images	3,614	454	456
Sentences	40,315	4,953	5,180
Relation instances	38,617	4,826	4,963
Attribute instances	37,580	4,644	4,588

Table 2: Aggregate statistics of the training, development (dev) and test sets.

Nearest neighbor Our first baseline computes a term-frequency vector for an input sentence and returns the scene graph of the nearest neighbor in the training data.

Object only Our second baseline is a parser that only outputs objects but no attributes or relationships. It uses the first two components of the classifier-based parser, namely the semantic graph processor and the object extractor, and then simply outputs all candidate objects.

We use the downstream performance on the image retrieval task as our main evaluation metric. We train our reimplementation of the model by Johnson et al. (2015) on our training set with human-constructed scene graphs. For each sentence we use the parser's output as a query and rank all images in the test set. For evaluation, we consider the human-constructed scene graph G_h of the sentence and construct a set of images $I = i_1, ..., i_n$ such that G_h is a subgraph of the image's complete scene graph. We compute the rank of each image in I and compute recall at 5 and 10 based on these ranks[3]. We also compute the median rank of the first correct result. We compare these numbers against an oracle system which uses the human-constructed scene graphs as queries instead of the scene graphs generated by the parser.

One drawback of evaluating on a downstream task is that evaluation is typically slower compared to using an intrinsic metric. We therefore also compare the parsed scene graphs to the human-constructed scene graphs. As scene graphs consist of object instances, attributes, and relations and are therefore similar to Abstract Meaning Representation (AMR) (Banarescu et al., 2013) graphs, we use Smatch F1 (Cai and Knight, 2013) as an additional intrinsic metric.

[3] As in Johnson et al. (2015), we observed that the results for recall at 1 were very unstable so we only report recall at 5 and 10 which are typically also more relevant for real-world systems that return multiple results.

	Development set				Test set			
	Smatch	**R@5**	**R@10**	**Med. rank**	**Smatch**	**R@5**	**R@10**	**Med. rank**
Nearest neighbor	32%	1.2%	2.3%	206	32%	1.1%	2.3%	205
Object only	**48%**	15.0%	29.3%	20	**48%**	12.6%	24.8%	25
Rule	43%	16.4%	31.6%	17	44%	13.5%	**27.1%**	**20**
Classifier	47%	**16.7%**	**32.9%**	**16**	47%	**13.8%**	**27.1%**	**20**
Oracle	-	19.4%	39.8%	13	-	16.6%	33.4%	15

Table 3: Intrinsic (Smatch F1) and extrinsic (recall at 5 and 10, and median rank) performance of our two baselines, our rule-based and our classifier-based parser.

	R@5	**R@10**	**Med. rank**
Johnson et al. (2015)	30.3%	47.9%	11
Our implementation	27.6%	45.6%	12

Table 4: Comparison of the results of the original implementation by Johnson et al. (2015) and our implementation. Both systems were trained and tested on the data sets of the original authors.

5 Results and Discussion

Table 3 shows the performance of our baselines and our two final parsers on the development and held-out test set.

Oracle results Compared to the results of Johnson et al. (2015), the results of our oracle systems are significantly worse. To verify the correctness of our implementation, the original authors provided us with their training and test set. Table 4 shows that our reimplementation performs almost as well as their original implementation. We hypothesize that there are two main reasons for the drop in performance when we train and evaluate our system on our dataset. First, our dataset is a lot more diverse and contains many more object classes and relation and attribute types. Second, the original authors only use the most common queries for which there exist at least five results to retrieve images while we evaluate on all queries.

Effectiveness of Smatch F1 As mentioned in the previous section, having an intrinsic evaluation metric can reduce the length of development cycles compared to using only an extrinsic evaluation. We hoped that Smatch F1 would be an appropriate metric for our task but our results indicate that there is no strong correlation between Smatch F1 and the performance of the downstream task.

Comparison of rule-based and classifier-based system In terms of image retrieval performance, there does not seem to be a significant difference between our rule-based system and our classifier-based system. On the development set the classifier-based system slightly outperforms the rule-based system but on the test set both seem to work equally well. Nevertheless, their results differ in some cases. One strength of the classifier-based system is that it learns that some adjectival modifiers like *several* should not be attributes. It is also able to learn some basic implications such as *the shirt looks dirty* implies in the context of an image that the shirt is dirty. On the other hand, the rule-based system tends to be more stable in terms of extracting relations while the classifier-based system more often only extracts objects from a sentence.

Comparison to baselines As shown in Table 3, both of our parsers outperform all our baselines in terms of recall at 5 and 10, and the median rank. This difference is particularly significant compared to the *nearest neighbor* baseline which confirms the complexity of our dataset and shows that it is not sufficient to simply memorize the training data.

The *object only* baseline is a lot stronger but still performs consistently worse than our two parsers. To understand in what ways our parsers are superior to the *object only* baseline, we performed a qualitative analysis. A comparison of the results reveals that the image retrieval model is able to make use of the extracted relations and attributes. Figure 3 shows the top 5 results of our classifier-based parser and the *object only* baseline for the query *"The white plane has one blue stripe and one red stripe"*. While the *object only* model seems to be mainly concerned with finding good matches for the two *stripe* objects, the output of our parser successfully captures the relation between the plane and the stripes and correctly ranks the two planes with the blue and red stripes as the

Figure 3: Top 5 results of the object only baseline (top row) and our classifier-based parser (bottom row) for the query *"The white plane has one blue stripe and one red stripe"*. The *object only* system seems to be mainly concerned with finding images that contain two *stripe* objects at the expense of finding an actual plane. Our classifier-based parser also outputs the relation between the stripes and the plane and the colors of the stripes which helps the image retrieval system to return the correct results.

Figure 4: 3D scenes for the sentences *"There is a wooden desk with a red and green lamp on it"* and *"There is a desk with a notepad on it"*.

top results.

Error analysis The performance of both of our parsers comes close to the performance of the oracle system but nevertheless there still remains a consistent gap. One of the reasons for the lower performance is that some human-constructed scene graphs contain information which is not present in the description. The human annotators saw both the description and the image and could therefore generate scene graphs with additional information.

Apart from that, we find that many errors occur with sentences which require some external knowledge. For example, our parser is not able to infer that *"a woman in black"* means that a woman is wearing black clothes. Likewise it is not able to infer that *"a jockey is wearing a green shirt and matching helmet"* implies that he is wearing a green helmet.

Other errors occur in some sentences which talk about textures. For example, our parsers assume that "a dress with polka dots" implies that there is a relation between one *dress* object and multiple *polka dot* objects instead of inferring that there is one *dress* object with the attribute *polka-dotted*.

One further source of errors are wrong dependency parses. Both of our parsers heavily rely on correct dependency parses and while making the parser's training data more diverse did improve results, we still observe some cases where sentences are parsed incorrectly leading to incorrect scene graphs.

6 Other Tasks

As mentioned before, one appeal of parsing sentences to an intermediate representation is that we can also use our parser for other tasks that make use of similar representations. One of these tasks is generating 3D scenes from textual descriptions (Chang et al., 2014). Without performing any further modifications, we replaced their parser with our classifier-based parser and used the resulting system to generate 3D scenes from several indoor scene descriptions. Two of these generated scenes are shown in Figure 4. Our impression is that the system performs roughly equally well using this parser compared to the one used in the original work.

7 Related Work

Image retrieval Image retrieval is one of the most active areas in computer vision research. Very early work mainly focused on retrieving images based on textual descriptions, while later work focused more on content-based image retrieval systems which perform retrieval directly based on image features. Rui et al. (1999), Liu et al. (2007), and Siddiquie et al. (2011) provide overviews of the developments of this field over the last twenty years. Most of this work focused on retrieving images from keywords which are not able to capture many semantic phenomena as well as natural language or our scene graph representation can.

Multi-modal embeddings Recently, multi-modal embeddings of natural language and images got a lot of attention (Socher et al., 2014; Karpathy et al., 2014; Plummer et al., 2015; Kiros et al., 2015; Mao et al., 2015; Chrupala et al., 2015). These embeddings can be used to retrieve images from captions and generating captions from images. As mentioned in the introduction, these models are trained on single-sentence image descriptions which typically cannot capture all the details of a visual scene. Further, unlike our modular system, they cannot be used for other tasks that require an interpretable semantic representation.

Parsing to graph-based representations Representing semantic information with graphs has recently experienced a resurgence caused by the development of the Abstract Meaning Representation (AMR) (Banarescu et al., 2013) which was followed by several works on parsing natural language sentences to AMR (Flanigan et al., 2014; Wang et al., 2015; Werling et al., 2015). Considering that AMR graphs are, like dependency trees, very similar to scene graphs, we could have also used this representation and transformed it to scene graphs. However, the performance of AMR parsers is still not competitive with the performance of dependency parsers which makes dependency trees are more stable starting point.

There also exists some prior work on parsing scene descriptions to semantic representations. As mentioned above, Chang et al. (2014) present a rule-based system to parse natural language descriptions to *scene templates*, a similar graph-based semantic representation. Elliott et al. (2014) parse image descriptions to a dependency grammar representation which they also use for image retrieval. Lin et al. (2014a) also use rules to transform dependency trees into semantic graphs which they use for video search. All of this work, however, only consider a limited set of relations while our approach can learn an arbitrary number of relations. Further, they all exclusively use very specific rule-based systems whereas we also introduced a more general purposed classifier-based parser.

8 Conclusion

We presented two parsers which can translate image descriptions to scene graphs. We showed that their output is almost as effective for retrieving images as human-generated scene graphs and that including relations and attributes in queries outperforms a model which only considers objects. We also demonstrated that our parser is well suited for other tasks which require a semantic representation of a visual scene.

Acknowledgments

We thank the anonymous reviewers for their thoughtful feedback. This work was supported in part by a gift from IPSoft, Inc. and in part by the Defense Advanced Research Projects Agency (DARPA) Broad Operational Language Translation (BOLT) program through IBM. The second author is also supported by a Magic Grant from The Brown Institute for Media Innovation. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the view of either DARPA or the US government.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*.

Shu Cai and Kevin Knight. 2013. Smatch: An evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association of for Computational Linguistics*.

Angel X Chang, Manolis Savva, and Christopher D Manning. 2014. Learning spatial knowledge for

text to 3D scene generation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Grzegorz Chrupala, Akos Kadar, and Afra Alishahi. 2015. Learning language through pictures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Desmond Elliott, Victor Lavrenko, and Frank Keller. 2014. Query-by-example image retrieval using visual dependency representations. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Jerry R Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Andrej Karpathy, Armand Joulin, and Fei Fei F Li. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. 2015. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics*.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. 2014a. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014b. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014*. Springer.

Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. 2007. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Heng Huangzhi, and Alan Yuille. 2015. Deep captioning with multimodal recurrent neural networks (m-RNN). In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The Penn treebank. *Computational linguistics*, 19(2):313–330.

Rick Nouwen. 2015. Plurality. In Paul Dekker and Maria Aloni, editors, *Cambridge Handbook of Semantics*. Cambridge University Press, Cambridge.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Bryan Plummer, Liwei Wang, Chris Cervantes, Juan Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *arXiv preprint arXiv:1505.04870*.

Yong Rui, Thomas S Huang, and Shih-Fu Chang. 1999. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62.

Behjat Siddiquie, Rogerio S Feris, and Larry S Davis. 2011. Image ranking and retrieval based on multi-attribute queries. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Raffaele Simone and Francesca Masini. 2014. On light nouns. *Word Classes: Nature, typology and representations*, 332:51.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2015. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*.

Chuan Wang, Nianwen Xue, Sameer Pradhan, and Sameer Pradhan. 2015. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT 2015)*.

Keenon Werling, Gabor Angeli, and Christopher D. Manning. 2015. Robust subgraph generation improves abstract meaning representation parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

C Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. 2013. Learning the visual interpretation of sentences. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 1681–1688. IEEE.