Visual Madlibs: Fill in the blank Description Generation and Question Answering

Licheng Yu, Eunbyung Park, Alexander C. Berg, Tamara L. Berg
Department of Computer Science, University of North Carolina, Chapel Hill
{licheng, eunbyung, aberg, tlberg}@cs.unc.edu

Abstract

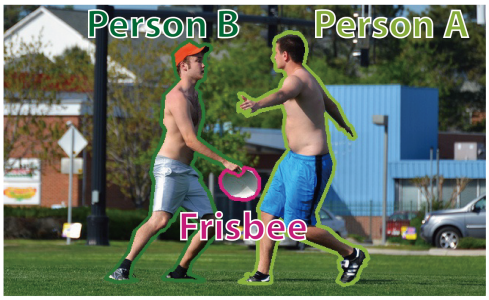
In this paper, we introduce a new dataset consisting of 360,001 focused natural language descriptions for 10,738 images. This dataset, the Visual Madlibs dataset, is collected using automatically produced fill-in-the-blank templates designed to gather targeted descriptions about: people and objects, their appearances, activities, and interactions, as well as inferences about the general scene or its broader context. We provide several analyses of the Visual Madlibs dataset and demonstrate its applicability to two new description generation tasks: focused description generation, and multiple-choice question-answering for images. Experiments using joint-embedding and deep learning methods show promising results on these tasks.

1. Introduction

Much of everyday language and discourse concerns the visual world around us, making understanding the relationship between the physical world and language describing that world an important challenge problem for AI. Understanding this complex and subtle relationship will have broad applicability toward inferring human-like understanding for images, producing natural human robot interactions, and for tasks like natural language grounding in NLP. In computer vision, along with improvements in deep learning based visual recognition, there has been an explosion of recent interest in methods to automatically generate natural language descriptions for images [6, 10, 16, 35, 17, 22] or videos [34, 9]. However, most of these methods and existing datasets have focused on only one type of description, a generic description for the entire image.

In this paper, we collect a new dataset of focused, targeted, descriptions, the *Visual Madlibs dataset*[1], as illustrated in Figure 1. To collect this dataset, we introduce automatically produced fill-in-the-blank templates designed to collect a range of different descriptions for visual content in an image. This is inspired by Madlibs, a childrens' word game where one player prompts another for a list of words to substitute for blanks in a story. In our case, a user might be presented with an image and a fill-in-the-blank template such as "The frisbee is [blank]" and asked to fill in the [blank] with a description of the appearance of frisbee. Alternatively, they could be asked to fill in the [blank] with a description of what the person is doing with the frisbee. Fill-in-the-blank questions can be targeted to collect descriptions about people and objects, their appearances, activities, and interactions, as well as descriptions of the general scene or the broader emotional, spatial, or temporal context of an image (examples in Fig 2). Using these templates, we collect 360,001 targeted descriptions for 10,738 images from the MS COCO collection [23].

With this new dataset, we can develop methods to generate more focused descriptions. Instead of asking an algo-



1. This place is a park.
2. When I look at this picture, I feel competitive.
3. The most interesting aspect of this picture is the guys playing shirtless.
4. One or two seconds before this picture was taken, the person caught the frisbee.
5. One or two seconds after this picture was taken, the guy will throw the frisbee.
6. Person A is wearing blue shorts.
7. Person A is in front of person B.
8. Person A is blocking person B.
9. Person B is a young man wearing an orange hat.
10. Person B is on a grassy field.
11. Person B is holding a frisbee.
12. The frisbee is white and round.
13. The frisbee is in the hand of the man with the orange cap.
14. People could throw the frisbee.
15. The people are playing with the frisbee.

Figure 1. An example from the Visual Madlibs Dataset, including a variety of targeted descriptions for people and objects.

[1] http://tamaraberg.com/visualmadlibs/

Figure 2. Example Visual Madlibs fill-in-the-blank descriptions.

rithm to "describe the image" we can now ask for more focused descriptions such as "describe the person", "describe what the person is doing," or "describe the relationship between the person and the frisbee." We can also ask questions about aspects of an image that are somewhat beyond the scope of the directly depicted content. For example, "describe what might have happened just before this picture was taken." or "describe how this image makes you feel." These types of descriptions reach toward high-level goals of producing human-like visual interpretations for images.

In addition to focused description generation, we also introduce a multiple-choice question-answering task for images. In this task, the computer is provided with an image and a partial description such as "The person is [blank]". A set of possible answers is also provided, one answer that was written about the image in question, and several additional answers written about other images. The computer is evaluated on how well it can select the correct choice. In this way, we can evaluate performance of description generation on a concrete task, making evaluation more straightforward. Varying the difficulty of the negative answers—adjusting how similar they are to the correct answer—provides a nuanced measurement of performance.

For both the generation and question-answering tasks, we study and evaluate a recent state of the art approach for image description generation [35], as well as a simple joint-embedding method learned on deep representations. The evaluation also includes extensive analysis of the Visual Madlibs dataset and comparisons to the existing MS COCO dataset of natural language descriptions for images.

In summary, our contributions are:
1) A new description collection strategy, *Visual Madlibs*, for constructing fill-in-the-blank templates to collect targeted natural language descriptions.
2) A new Visual Madlibs Dataset consisting of 360,001 targeted descriptions, spanning 12 different types of templates, for 10,738 images, as well as analysis of the dataset and comparisons to existing MS COCO descriptions.
3) Evaluation of a generation method and a simple joint embedding method for targeted description generation.
4) Definition and evaluation of generation and joint-embedding methods on a new task, *multiple-choice fill-in-the-blank question answering for images*.

The rest of our paper is organized as follows. First, we review related work (Sec 2). Then, we describe our strategy for automatically generating fill-in-the-blank templates and introduce our Visual Madlibs dataset (Sec 3). Next we outline the multiple-choice question answering and targeted generation tasks (Sec 4) and provide several analyses of our dataset (Sec 5). Finally, we provide experiments evaluating description generation and joint-embedding methods on the proposed tasks (Sec 6) and conclude (Sec 7).

2. Related work

Description Generation: Recently, there has been an explosion of interest in methods for producing natural language descriptions for images or video. Early work in this area focused on detecting content elements and then composing captions [20, 36, 28, 11, 18] or made use of existing text either directly associated with an image [12, 1] or retrieved from visually similar images [29, 21, 26]. With the advancement of deep learning for content estimation, there have been many exciting recent attempts to generate image descriptions using neural network based approaches. Some methods first detect words or phrases using Convolutional Neural Network (CNN) features, then generate and re-rank candidate sentences [10, 22]. Other approaches take a more end-to-end approach to generate output descriptions directly from images [17, 35, 16, 6]. These new methods have shown great promise for image description generation, under some measures (e.g. BLEU-1) achieving near-human performance levels.

Description Datasets: Along with the development of image captioning algorithms there have been a number of datasets collected for this task. One of the first datasets collected for this problem was the UIUC Pascal Sentence data set [11] which contains 1,000 images with 5 sentences per image written by workers on Amazon Mechanical Turk. Based on this, PASCAL-50s [33] further collected 50 sentences per image. As the description problem gained popularity larger and richer datasets were collected, including

the Flickr8K [30] and Flickr30K [37] datasets. In an alternative approach, the SBU Captioned photo dataset [29] contains 1 million images with existing captions collected from Flickr, but the text tends to contain more contextual information since captions were written by the photo owners. Most recently, Microsoft released the MS COCO [23] dataset, containing 120,000 images depicting 80 common object classes, with object segmentations and 5 turker written descriptions per image. We make use of MS COCO, extending the types of descriptions associated with images.

Question-answering Natural language question-answering has been a long standing goal of NLP, with commercial companies like Ask-Jeeves or Google playing a significant role in developing effective methods. Recently, embedding and deep learning methods have shown great promise [32, 3, 4]. Lin *et al.* [24] take an interesting multi-modal approach to question-answering. A multiple-choice text-based question is first constructed from 3 sentences written about an image; 2 of the sentences are used as the question, and 1 is used as the positive answer, mixed with several negative answers from sentences written about other images. The authors develop ranking methods to answer these questions and show that generating abstract images for each potential answer can improve results. Note, here the algorithms are not provided with an image as part of the question. Some recent work has started to look at the problem of question-answering for images. Malinowski *et al.* [25] introduced two scene-based QA datasets and combine computer vision and NLP in a Bayesian framework. DAQUAR is made by collecting human questions and answers, and SynthQA is automatically generated based on object segmentation and question templates. Geman *et al.* [13] design a visual Turing test to evaluate image understanding using a series of binary questions about image content. We design question-answering tasks that are somewhat broader in scope than the previous works, allowing us to ask a variety of different types of natural language questions about images.

3. Designing and collecting Visual Madlibs

The goal of Visual Madlibs is to study targeted natural language descriptions of image content that go beyond generic descriptions of the whole image. The experiments in this paper begin with a dataset of images where the presence of some objects have already been labeled. The prompts for the questions are automatically generated based on image content, in a manner designed to elicit more detailed descriptions of the objects, their interactions, and the broader context of the scene shown in each image.

Visual Madlibs: Image+Instruction+Prompts+Blank
A single fill-in-the-blank question consists of a prompt and a blank, e.g., *Person A is [blank] the car.* The implicit question is, "What goes in the blank?" This is presented to a person along with an image and instructions, e.g., *Describe the relationship between the indicated person and object.* The same image and prompt may be used with different instructions to collect a variety of description types.

Instantiating Questions
While the general form of the questions for the Visual Madlibs were chosen by hand, see Table 1, most of the questions are instantiated depending on a subset of the objects present in an image. For instance, if an image contained two people and a dog, questions about each person (question types 9-11 in Table 1), the dog (types 6-8), relationships between the two people and the dog (type 12), could be instantiated. For each possible instantiation, the wording of the questions will be automatically altered slightly to maintain grammatical consistency. In addition to these types of questions, other questions (types 1-5) can be instantiated for an image regardless of the objects present.

Notice in particular the questions about the temporal context – what might have happened before or what might happen after the image was taken. People can make inferences beyond the specific content depicted in an image. Sometimes these inferences will be consistent between people (e.g., when what will happen next is obvious), and other times these descriptions may be less consistent. We can use the variability of returned responses to select images for which these inferences are reliable.

Asking questions about every object and all pairs of objects quickly becomes unwieldy as the number of objects increases. To combat this, we choose a subset of objects present to use in instantiating questions. Such selection could be driven by a number of factors. The experiments in this paper consider comparisons to existing, general, descriptions of images, so we instantiate questions about the objects mentioned in those existing natural language descriptions, an indication of the object's importance [2].

3.1. Data Collection

To collect the Visual Madlibs Dataset we use a subset of 10,738 human-centric images from MS COCO, that make up about a quarter of the validation data [23], and instantiate fill-in-the-blank templates as described above. The MS COCO images are annotated with a list of objects present in the images, segmentations for the locations of those objects, and 5 general natural language descriptions of the image. To select the subset of images for collecting Madlibs, we start with the 19,338 images with a person labeled. We then look at the five descriptions for each and perform a dependency parse [8], only keeping those images where a word referring to person is the head noun of the parse. This leaves 14,150 images. We then filter out the images whose descriptions do not include a synonym for any of the 79 non-person object categories labeled in the MS COCO dataset. This leaves 10,738 human-centric images with at least one other object from the MS COCO data set mentioned in the

Type	Instruction	Prompt	#words
1. image's scene	Describe the type of scene/place shown in this picture.	The place is a(n) ____ .	4+1.45
2. image's emotion	Describe the emotional content of this picture.	When I look at this picture, I feel ____ .	8+1.14
3. image's interesting	Describe the most interesting or unusual aspect of this picture.	The most interesting aspect of this picture is ____ .	8+3.14
4. image's past	Describe what happened immediately before this picture was taken.	One or two seconds before this picture was taken, ____ .	9+5.45
5. image's future	Describe what happened immediately after this picture was taken.	One or two seconds after this picture was taken, ____ .	9+5.04
6. object's attribute	Describe the appearance of the indicated object.	The object(s) is/are ____ .	3.20+1.62
7. object's affordance	Describe the function of the indicated object.	People could ____ the object(s).	4.20+1.74
8. object's position	Describe the position of the indicated object.	The object(s) is/are ____ .	3.20+3.35
9. person's attribute	Describe the appearance of the indicated person/people.	The person/people is/are ____ .	3+2.52
10. person's activity	Describe the activity of the indicated person/people.	The person/people is/are ____ .	3+2.47
11. person's location	Describe the location of the indicated person/people.	The person/people is/are ____ .	3.20+3.04
12. pair's relationship	Describe the relationship between the indicated person and object.	The person/people is/are ____ the object(s).	5.20+1.65

Table 1. All 12 types of Madlibs instructions and prompts. Right-most column shows the average number of words for each description (#words for prompt + #words for answer).

general image descriptions. Before final instantiation of the fill-in-the blank templates, we need to resolve a potential ambiguity regarding which objects are referred to in the descriptions. We would like to collect Madlibs for objects described in the MS COCO captions, but since correspondences between the segmented objects and description mentions are not available, we first try to automatically estimate this assignment by parsing the descriptions. We consider two possible cases: 1) there are fewer annotated instances than the sentences describe, 2) there are more annotated instances than the sentences describe. It is easy to address the first case, just construct templates for all of the labeled instances. For the second case, we sort the area of each segmented instance, and pick the largest ones up to the parsed number for instantiation. Using this procedure, we obtain 26,148 labeled object or person instances in 10,738 images.

Each Visual Madlib is answered by 3 workers on Amazon Mechanical Turk. To date, we have collected 360,001 answers to Madlib questions and are continuing collection to include the training portion of the MS COCO dataset.

4. Tasks: Multiple-choice question answering and targeted generation

We design two tasks to evaluate targeted natural language description for images. The first task is to automatically generate natural language descriptions of images to fill in the blank for one of the Madlibs questions. The input to this task is an image, instructions, and a Madlibs prompt. As has been discussed in the community working on description generation for images, it can be difficult to evaluate free form generation [33]. Our second task tries to address this issue by developing a new targeted multiple-choice question answering task for images. Here the input is again an image, instruction, and a prompt, but instead of a free form text answer, there are a fixed set of multiple-choice answers to fill in the blank. The possible multiple-choice answers are sampled from the Madlibs responses, one that was written for the particular image/instruction/prompt as the correct answer, and distractors chosen from either similar images or random images depending on the level of difficulty desired. This ability to choose distractors to adjust the difficulty of the question as well as the relative ease of evaluating multiple choice answers are attractive aspects of this new task.

In our experiments we randomly select 20% of the 10,738 images to use as our test set for evaluating these tasks. For the multiple-choice questions we form two sets of answers for each, with one set designed to be more difficult than the other. We first establish the easy task distractor answers by randomly choosing three descriptions (of the same question type) from other images [24]. The hard task is designed more delicately. Instead of randomly choosing from the other images, we now only look for those containing the same objects as our question image, and then arbitrarily pick three of their descriptions. Sometimes, the descriptions sampled from "similar" images could also be good answers for our questions (later we experiment with using Turkers to select less ambiguous multiple-choice questions from this set). For the targeted generation task, for question types 1-5, algorithms generate descriptions given the image, instructions, and prompt. For the other question types whose prompts are related to some specific person or object, we additionally provide the algorithm with the location of each person/object mentioned in the prompt. We also experiment with estimating these locations using object detectors.

5. Analyzing the Visual Madlibs Dataset

We begin by conducting quantitative analyses of the responses collected in the Visual Madlibs Dataset in Sec. 5.1. A main goal is understanding what additional information is provided by the targeted descriptions in the Visual Madlibs Dataset vs general image descriptions. Therefore, we also provide analyses comparing Visual Madlibs to MS COCO descriptions collected for the same images in Sec. 5.2.

5.1. Quantifying Visual Madlibs responses

We analyze the length, structure, and consistency of the Visual Madlibs responses. First, the average length of each type of description is shown in the far right column of Table 1. Note that descriptions of people tend to be longer than descriptions of other objects in the dataset.

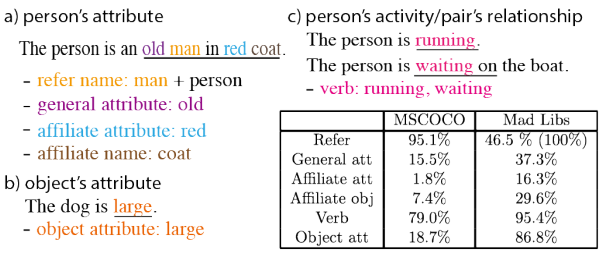
Second, we use phrase chunking [7] to analyze which phrasal structures are commonly used to fill in the blanks for different questions. Fig. 3, top row, shows relative frequencies for the top-5 most frequent templates used for several question types. Object attributes are usually described briefly with a simple adjectival phrase. On the other hand, people use more words and a wider variety of structures to describe possible future events. Except for future and past descriptions, the distribution of structures is generally concentrated on a few likely choices for each question type.

Third, we analyze how consistent the Mechanical Turk workers' answers are for each type of question. To compute a measure of similarity between a pair of responses we use the cosine similarity between representations of each response. A response is represented by the mean of the Word2Vec [27] vectors for each word in the response, following [24, 22]. Word2Vec is a 300 dimensional embedding representation for words that encodes the distributional context of words learned over very large word corpora. This measure takes into account the actual words used in a response, as opposed to the previous analyses of parse structure. Each Visual Madlibs question is answered by three workers, providing 3 pairs for which similarity is computed. Fig. 3, bottom row, shows a histogram of all pairwise similarities for several question types. Generally the similarities have a normal-like distribution with an extra peak around 1 indicating the fraction of responses that agree almost perfectly. Once again, descriptions of the future and past are least likely to be (near) identical, while object attributes and affordances are often very consistent.

5.2. Visual Madlibs vs general descriptions

We compare the targeted descriptions in the Visual Madlibs Dataset to the general image descriptions in MS COCO. First, we analyze the words used in Visual Madlibs compared to MS COCO descriptions of the same images. For each image, we extract the unique set of words from all descriptions of that image from both datasets, and compute the coverage of each set with respect to the other. We find that on average (across images) 22.45% of the Madlibs's words are also present in MS COCO descriptions, while 52.38% of the MS COCO words are also present in Madlibs. We also compute the vocabulary size of Madlibs that is 12,329, compared with MS COCO's 9,683 on the same image set.

Second, we compare how Madlibs and MS COCO answers describe the people and objects in images. We observe that the Madlibs questions types, Table 1, cover much of the information in MS COCO descriptions [22]. As one way to see this, we run the StanfordNLP parser on both datasets [5]. For attributes of people, we use the parsing template shown in Fig. 4(a) to analyze the structures being used. The *refer name* indicates whether the person was mentioned in the description. Note that the Madlibs descrip-



	MSCOCO	Mad Libs
Refer	95.1%	46.5 % (100%)
General att	15.5%	37.3%
Affiliate att	1.8%	16.3%
Affiliate obj	7.4%	29.6%
Verb	79.0%	95.4%
Object att	18.7%	86.8%

Figure 4. Template used for parsing person's attributes, activity and interaction with object, and object's attribute. The percentages below compares Madlibs and MS COCO on how frequent these templates are used for description.

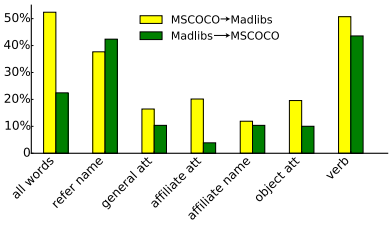


Figure 5. Frequency that a word in a position in the people and object parsing template in one dataset is in the same position for the other dataset.

tions always have one reference to a person in the prompt (The person is [blank].). Therefore, for Madlibs, we report the presence of additional references to the person (e.g., the person is a *man*). The *general attribute* directly describes the appearance of the person or object (e.g., old or small); the *affiliate object* indicates whether additional objects are used to describe the targeted person (e.g. with a bag, coat, or glasses) and the *affiliate attribute* are appearance characteristics of those secondary objects (e.g., red coat). The templates for *object's attribute* and *verbs* are more straightforward as shown in Fig. 4(b)(c). The table in Fig. 4 shows the frequency of each parse component. Overall, more of the potential descriptive elements in these constructions are used in response to the Madlibs prompts than in the general descriptions found in MS COCO.

We also break down the overlap between Visual Madlibs and MS COCO descriptions over different parsing templates for descriptions about people and object (Fig. 5). Yellow bars show how often words for each parse type in MS COCO descriptions were also found in the same parse type in the Visual Madlibs answers, and green bars measure the reverse direction. Observations indicate that Madlibs provides more coverage in its descriptions than MS COCO for all templates except for person's refer name. One possible reason is that the prompts already indicates "the person" or "people" explicitly, so workers need not add an additional reference to the person in their descriptions.

Extrinsic comparison of Visual Madlibs Data and general descriptions: We perform an extrinsic analysis by us-

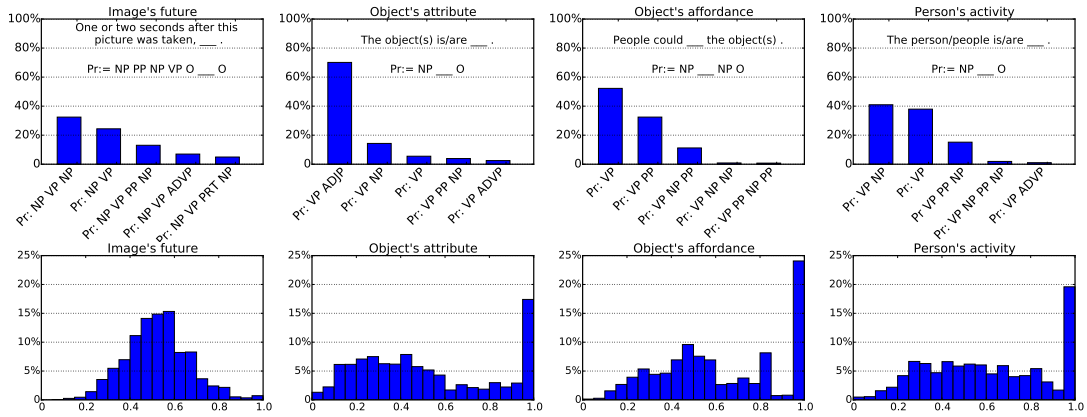


Figure 3. First row shows top-5 most frequent phrase templates for image's future, object's attribute, object's affordance and person's activity. Second row shows the histograms of similarity between answers. (We put the plots for all 12 types in the supplementary file.)

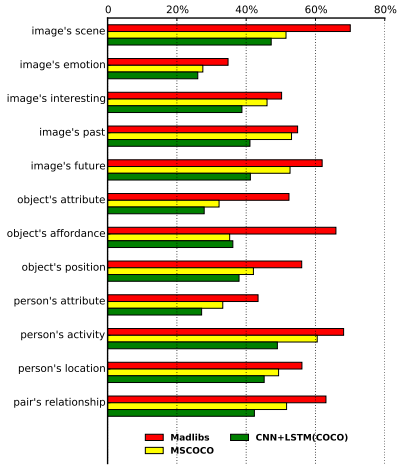


Figure 6. The accuracy of Madlibs, MS COCO and CNN+LSTM [35] (trained on MS COCO) used as references to answer the Madlibs hard multiple-choice questions.

ing either: a) the MS COCO descriptions for an image, or b) Visual Madlibs responses from other Turkers for an image, to select answers for our multiple-choice evaluation task. Specifically, we use one of the human provided descriptions, and select the multiple-choice answer that is most similar to that description. Similarity is measured as cosine similarity between the mean Word2Vec vectors for the words a description compared to the multiple-choice answers. In addition to comparing how well the Madlibs or MS COCO descriptions can select the correct multiple-choice answer, we also use the descriptions automatically produced by a recent CNN+LSTM description generation system [35][2]. trained on MS COCO dataset. This allows us to make one possible measurement of how close current automatically generated image descriptions are to our Madlibs descriptions. Fig. 6 shows the accuracies resulting from using Madlibs, MS COCO, or CNN+LSTM [35] to select the correct multiple-choice answer.

Although this approach is quite simple, it allows us we make two interesting observations. First, Madlibs outperforms MS COCO on all types of multiple-choice questions. If Madlibs and MS COCO descriptions provided the same information, we would expect their performance to be comparable. Second, the automatically generated descriptions from the pre-trained CNN+LSTM perform much worse than the actual MS COCO descriptions, despite doing well on general image description generation.

6. Experiments

In this section we evaluate a series of methods on the targeted natural language generation and multiple-choice question answering tasks. As methods, we evaluate a language only baseline, which computes the 4-gram perplexity for each sentence using Google-1T statistics (frequencies of all n-grams on the web). We also try simple joint-embedding methods – canonical correlation analysis (CCA) and normalized CCA (nCCA) [15] – as well as a recent deep-learning based method for image description generation CNN+LSTM [35]. We train these models on 80% of the images in the MadLibs collection and evaluate their performance on the remaining 20%.

In our experiments we extract image features using the VGG Convolutional Neural Network (VGGNet) [31], trained on the ILSVRC-2012 dataset to recognize 1000 object classes.For comparison, we also extract image features using the Places-CNN, which is trained on 205 scene categories of Places Database [38] using AlexNet [19]. On the sentence side, we average the Word2Vec of all words in a sentence to obtain a representation.

CCA finds a joint embedding between two multi-

[2]In this paper, we use Karpathy's implementation: `https://github.com/karpathy/neuraltalk`

Easy Task

	#Q	n-gram	CCA	nCCA	nCCA (place)	nCCA (bbox)	nCCA (all)	CNN+LSTM (madlibs)	CNN+LSTM(r) (madlibs)	Human
1. scene	6277	24.8%	75.7%	86.8%	85.4%	–	**87.6%**	74.2%	77.6%	93.2%
2. emotion	5138	26.7%	41.3%	49.2%	**50.4%**	–	42.4%	37.0%	44.5%	48.3%
3. past	4903	24.3%	61.8%	77.5%	72.6%	–	**80.3%**	50.1%	47.3%	93.5%
4. future	4658	27.7%	61.2%	78.0%	72.1%	–	**80.2%**	50.6%	50.6%	94.5%
5. interesting	5095	24.2%	66.8%	76.5%	72.0%	–	**78.9%**	55.4%	49.9%	94.7%
6. obj attr	7194	30.6%	44.1%	47.5%	44.7%	54.7%	50.9%	46.9%	**59.0%**	88.9%
7. obj aff	7326	30.1%	59.8%	73.0%	69.6%	72.2%	76.7%	–	**88.9%**	93.1%
8. obj pos	7290	28.0%	53.0%	65.9%	64.2%	58.9%	**69.7%**	53.9%	69.6%	91.4%
9. per attr	6651	27.2%	40.4%	48.0%	44.5%	**53.1%**	44.5%	36.5%	46.0%	83.8%
10. per act	6501	27.3%	70.0%	80.7%	76.9%	75.6%	**82.8%**	64.7%	68.9%	96.7%
11. per loc	6580	24.4%	69.8%	82.7%	82.6%	73.8%	**82.7%**	60.8%	71.6%	92.2%
12. pair rel	7595	29.2%	54.3%	63.0%	61.3%	64.2%	67.2%	–	**72.3%**	91.7%

Hard Task

	#Q	n-gram	CCA	nCCA	nCCA (place)	nCCA (bbox)	nCCA (all)	CNN+LSTM (madlibs)	CNN+LSTM(r) (madlibs)	Human
1. scene	6277	22.8%	63.8%	70.1%	**70.7%**	–	68.2%	63.6%	64.2%	75.6%
2. emotion	5138	25.1%	33.9%	37.2%	**38.3%**	–	33.2%	34.6%	37.6%	38.4%
3. past	4903	22.4%	47.9%	52.8%	49.5%	–	**54.0%**	42.2%	39.5%	73.9%
4. future	4658	24.4%	47.5%	**54.3%**	50.5%	–	53.3%	41.1%	39.5%	75.1%
5. interesting	5095	27.6%	51.4%	53.7%	50.5%	–	**55.1%**	44.0%	37.1%	76.7%
6. obj attr	7194	29.5%	42.2%	43.6%	41.5%	**49.8%**	39.3%	41.6%	42.3%	70.5%
7. obj aff	7326	32.2%	54.5%	63.5%	60.9%	63.0%	48.5%	–	**69.4%**	52.7%
8. obj pos	7290	29.2%	49.0%	**55.7%**	53.3%	50.7%	53.4%	46.7%	50.2%	70.8%
9. per attr	6651	23.3%	33.9%	38.6%	35.5%	**46.1%**	31.6%	35.5%	42.4%	70.5%
10. per act	6501	24.0%	59.7%	65.4%	62.6%	65.1%	**66.6%**	57.3%	53.7%	85.1%
11. per loc	6580	22.3%	56.8%	63.3%	**65.5%**	57.8%	62.6%	50.4%	56.8%	72.9%
12. pair rel	7595	30.1%	49.4%	54.3%	52.2%	**56.5%**	52.0%	–	54.6%	74.7%

Table 2. Accuracies computed for different approaches on the easy and hard multiple-choice answering task. CCA, nCCA, and CNN+LSTM are trained on the whole image representation for each type of question. nCCA(place) uses Places-CNN feature. nCCA(box) is trained and evaluated on ground-truth bounding-boxes from MS COCO segmentations. nCCA(all) trains a single embedding using all question types. CNN+LSTM(r) ranks the perplexity of {prompt+choice}.

Figure 7. Some hard multiple-choice question examples. The results are made by nCCA. First row shows correct choices. Second row shows incorrect choices. Corresponding human accuracies are provided as reference.

dimensional variables, in our case image and text vector representations. To increase the flexibility of the feature selection and for improving computational efficiency, Gong *et al.* [15] proposed nCCA a scalable approximation scheme of explicit kernel mapping followed by dimension reduction and linear CCA. In the projected latent space, the similarity is measured by the eigenvalue-weighted normalized correlation. We train CCA and nCCA models for each question type separately using the training portion of the Visual Madlibs Dataset. These models allow us to map from an image representation, to the joint-embedding space, to vectors in the Word2Vec space, and vice versa. For targeted generation, we map an image to the joint-embedding space and then choose the answer from the training set text that is

closest to this embedded point. To answer multiple-choice questions, we embed each multiple choice answer and then select the answer whose embedding is closest.

Following recent description generation techniques [35, 16], we train a CNN+LSTM model for each question type. These models learn a mapping from an image and prompt to a sequence of wordse.g., The chair is, and then let the CNN+LSTM system generate the remaining words of the description. For the multiple choice task, we evaluate two ways to select an answer. The first method selects the answer with largest cosine Word2Vec similarity to the generated description. The second method ranks the prompt+choices by perplexity and selects the best one.

6.1. Discussion of results

Table 2 shows accuracies of each algorithm on the easy and hard versions of the multiple-choice task[3] and Fig. 7 shows example correct and incorrect answer choices. There are several interesting observations we can make. From the results of the language only n-gram baseline, we conclude that answering Madlibs questions strongly requires visual information. Second, training nCCA on all types of questions together, nCCA(all), is helpful for the easy variant of the task, but less useful on the more fine-grained hard version of the task. Third, extracting visual features from the bounding box of the relevant person/object yields higher accuracy for predicting attributes, but not for other questions. Based on this finding, we evaluate answering the attribute question using automatic detection methods. The detectors are trained on ImageNet using R-CNN [14], covering 42 MS COCO categories. We observe similar performance between ground-truth and detected bounding boxes in Table 4. Fourth, we observe that the Places-CNN helps answer questions related to image's scene, person's location, and image's emotion.

As an additional experiment we ask 5 people to answer each multiple choice question. The last column of Table 2 shows human accuray as a reference. We further use human agreement to select a subset of the multiple-choice questions where at least 3 Turkers choose the correct answer. Results of the methods on this question subset are shown in Table 3, displaying similar patterns as the unfiltered set, with slightly higher accuracy.

Finally, Table 5 shows BLEU-1 and BLEU-2 scores for targeted generation. Although the CNN+LSTM models we trained on Madlibs were not quite as accurate as nCCA for selecting the correct multiple-choice answer, they did result in better, sometimes much better, accuracy (as measured by BLEU scores) for targeted generation.

[3]The missing entries for questions 7 and 12 are due to priming not being valid for questions with blanks in the middle of the sentence.

Filtered Questions from **Hard Task**

	#Q	nCCA	nCCA (place)	nCCA (bbox)	nCCA (all)	CNN+LSTM(r) (madlibs)
1. scene	4940	77.6%	**77.8**%	–	76.3%	69.7%
2. emotion	2052	49.0%	**49.5**%	–	43.8%	43.0%
3. past	3976	57.4%	53.8%	–	**59.4**%	41.3%
4. future	3820	**59.2**%	54.2%	–	58.3%	41.7%
5. interesting	4159	59.5%	55.1%	–	**61.3**%	40.3%
6. obj attr	5436	47.2%	44.7%	**54.6**%	42.8%	46.3%
7. obj aff	4581	71.0%	67.6%	70.5%	57.6%	**79.0**%
8. obj pos	5721	**60.2**%	57.7%	54.6%	57.7%	54.3%
9. per attr	4893	42.4%	38.8%	**52.1**%	34.4%	46.4%
10. per act	5813	68.3%	65.3%	67.9%	**69.6**%	55.3%
11. per loc	5096	69.9%	**71.7**%	62.6%	70.0%	60.6%
12. pair rel	5981	57.6%	55.4%	**60.0**%	56.5%	57.4%

Table 3. Accuracies for different approaches on the filtered questions from hard task. The filtered questions are those with human accuracies higher than 0.6. Full tables for filtered easy and hard task are in the supplementary file.

		Easy Task			**Hard Task**		
	#Q	nCCA	nCCA (bbox)	nCCA (dbox)	nCCA	nCCA (bbox)	nCCA (dbox)
6. obj attr	2021	47.6%	53.6%	51.4%	43.9%	47.9%	45.2%
9. per attr	4206	50.2%	55.4%	51.2%	40.0%	47.0%	43.3%

Table 4. Multiple-choice answering using automatic detection for 42 object/person categories. "bbox" denotes ground-truth bounding box and "dbox" denotes detected bounding box.

	BLEU-1			**BLEU-2**		
	nCCA	nCCA (box)	CNN+LSTM (madlibs)	nCCA	nCCA (bbox)	CNN+LSTM (madlibs)
1. scene	0.52	–	0.62	0.17	–	0.19
2. emotion	0.17	–	0.38	0	–	0
3. future	0.38	–	0.39	0.12	–	0.13
4. past	0.39	–	0.42	0.12	–	0.12
5. interesting	0.49	–	0.65	0.14	–	0.22
6. obj attr	0.28	0.36	0.48	0.02	0.02	0.01
7. obj aff	0.56	0.60	–	0.10	0.11	–
8. obj pos	0.53	0.55	0.71	0.24	0.25	0.49
9. per attr	0.26	0.29	0.57	0.06	0.07	0.25
10. per act	0.47	0.41	0.53	0.14	0.11	0.20
11. per loc	0.52	0.46	0.63	0.22	019	0.39
12. pair rel	0.46	0.48	–	0.07	0.08	–

Table 5. BLEU-1 and BLEU-2 computed on Madlibs testing dataset for different approaches.

7. Conclusions

We have introduced a new fill-in-the blank strategy for collecting targeted natural language descriptions. Our analyses show that these descriptions are usually more detailed than generic whole image descriptions. We also introduce a targeted natural language description generation task, and a multiple-choice question answering task, then train and evaluate joint-embedding and generation models. Data produced by this paper will be publicly released.

Acknowledgements: We thank the vision and language communities for feedback, especially J. Hockenmaier, K. Saenko, and J. Corso. This research is supported by NSF Awards #1417991, 1405822, 144234, 1452851, and Microsoft Research.

References

[1] A. Aker and R. Gaizauskas. Generating image descriptions using dependency relational patterns. In *ACL*, 2010.

[2] A. C. Berg, T. L. Berg, H. D. III, J. Dodge, A. Goyal, X. Han, A. Mensch, M. Mitchell, A. Sood, K. Stratos, and K. Yamaguchi. Understanding and predicting importance in images. In *CVPR*, 2012.

[3] A. Bordes, J. Weston, and S. Chopra. Question answering with subgraph embeddings. In *EMNLP*, 2014.

[4] A. Bordes, J. Weston, and N. Usunier. Open question answering with weakly supervised embedding models. In *ECML PKDD*, 2014.

[5] D. Chen and C. D. Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, 2014.

[6] X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, 2015.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *JMLR*, 2011.

[8] M.-C. De Marneffe, B. MacCartney, C. D. Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, 2006.

[9] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[10] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. Lawrence Zitnick, and G. Zweig. From captions to visual concepts and back. In *CVPR*, 2015.

[11] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010.

[12] Y. Feng and M. Lapata. Topic models for image annotation and text illustration. In *ACL*, 2010.

[13] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 2015.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[15] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014.

[16] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *TACL*, 2015.

[18] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. *NAACL HLT 2013*, page 10, 2013.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. In *CVPR*, 2011.

[21] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In *ACL*, 2012.

[22] P. O. Lebret Remi, Pinheiro and R. Collobert. Phrase-based captioning. In *ICML*, 2015.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[24] X. Lin and D. Parikh. Don't just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *CVPR*, 2015.

[25] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014.

[26] R. Mason. Domain-independent captioning of domain-specific images. In *NACCL-HLT*, 2013.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[28] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III. Midge: Generating image descriptions from computer vision detections. In *EACL*, 2012.

[29] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[30] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[32] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. Weakly supervised memory networks. *arXiv preprint arXiv:1503.08895*, 2015.

[33] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[34] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015.

[35] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[36] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[37] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *TACL*, 2014.

[38] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.