Natural language guided object retrieval in images

Ahmad Ostovar¹ · Suna Bensch¹ · Thomas Hellström¹

Received: 31 January 2020 / Accepted: 20 April 2021
© The Author(s) 2021

Abstract

The ability to understand the surrounding environment and being able to communicate with interacting humans are important functionalities for many automated systems where visual input (e.g., images, video) and natural language input (speech or text) have to be related to each other. Possible applications are automatic image caption generation, interactive surveillance systems, or human robot interaction. In this paper, we propose algorithms for automatic responses to natural language queries about an image. Our approach uses a predefined neural net for detection of bounding boxes and objects in images, spatial relations between bounding boxes are modeled with a neural net, the queries are analyzed with a syntactic parser, and algorithms to map natural language to properties in the images are introduced. The algorithms make use of semantic similarity and antonyms. We evaluate the performance of our approach with test users assessing the quality of our system's generated answers.

1 Introduction

In human–computer interaction and human–robot interaction, a computer's or robot's ability to communicate in natural language with interacting humans depends mainly on two factors: the system has to understand the surrounding environment and has to be able to communicate in natural language about the environment. An environment might be the real world, a video or an image and consists of visual percepts such as objects, spatial relations, colors, and actions. The process of relating linguistic symbols in natural language input (e.g., nouns, verbs) to visual percepts is called *language grounding*.

One example, of a scenario where language has to be grounded, is surveillance systems, where an operator may ask questions about what is happening in the video streams.

In this paper, we propose an architecture and algorithms for language grounding, given an image and a natural language text query. In particular, we describe our approach for ana-

✉ Suna Bensch
suna@cs.umu.se

Ahmad Ostovar
ahmado@cs.umu.se

Thomas Hellström
thomash@cs.umu.se

¹ Department of Computing Science, Umeå University, 901 87 Umeå, Sweden

lyzing and labeling visual percepts, a method for analyzing linguistic symbols and introduce algorithms that map linguistic symbols occurring in the text query into visual percepts.

We consider three types of text queries:

1. Attention queries (type 1), e.g., “find the person to the right of the monitor,”
2. Relation queries (type 2), e.g., “where is the person?”,
3. Identification queries (type 3), e.g., “what is to the right of the monitor?”

For attention queries, our system returns two bounding boxes (which are rectangular boxes around objects in an image). For relation queries our systems returns two bounding boxes, an object category, and a spatial relation. For identification queries, two bounding boxes and an object category are returned.

Our approach builds on usage of a predefined neural net for detection of bounding boxes and objects in images. Spatial relations between bounding boxes are modeled with a neural net, the text queries are analyzed with a syntactic parser, and algorithms using semantic similarity and antonyms output the result of language grounding.

The performance of the developed system was assessed by test users who reported how well the system’s generated answers matched their own view for a given set of images and questions.

The paper is structured as follows. In Sect. 2, we discuss existing work and approaches for language grounding in images. In Sect. 3, we give an overall description of the proposed solution, followed by separate sections on image analysis in Sect. 4, spatial relations analysis in Sect. 5, and analysis of queries in Sect. 6. Section 7 describes novel algorithms for language grounding, through which natural language words are mapped to objects and spatial relations in an image. We evaluate our approach in Sect. 8, present results in Sect. 9, and conclude the paper with a discussion of results and suggested future work in Sect. 11.

2 Related work

Language grounding is the process of connecting linguistic symbols (e.g., nouns, verbs, sentences) to visual percepts in an image or video (e.g., objects, spatial relations between objects, actions). In particular, given an image containing a set of visual percepts $V = \{v_1, v_2, \dots, v_k\}$ and a text query containing a set of linguistic symbols $T = \{t_1, t_2, \dots, t_p\}$, the language grounding problem is an assignment problem where linguistic symbols $t_i \in T$ should be assigned to visual percepts $v_j \in V$. In the literature the language grounding problem is investigated under different names and approaches such as object retrieval using language grounding, relationship detection and Visual Question Answering (VQA).

Methods for object retrieval using language grounding identify objects in an image based on a query text that includes properties of objects such as attributes, categories and spatial relations.

Addressing the similar problem as ours, Guadarrama et al. [11] developed a system to localize objects in images based on a text query by generating text as bag-of-words from candidate boxes using the class labels which are predicted from a pretrained object classifier, and then compares the text query and the bags. Ronghang et al. [15] developed a Spatial Context Recurrent ConvNet (SCRC) model as a scoring function for measuring the similarity of a text query and candidate boxes by integrating spatial configurations and global scene-level contextual information into the network. This work motivated our approach to use a similarity function to determine the analogy between the text query fragments, detected objects and their spatial relations.

Similarly, in [19], given an image and the text query, words in the query are aligned with image regions by embedding the detected objects as a result of a pretrained object detector and text fragments from a parser with a ranking loss. The work in [21] introduced Logical Semantics with Perception (LSP) model that learns to map natural language sequences to related objects in an image by grounding language acquisition. The authors in [33] developed an attention model which learns to ground phrases in images using the regions of an image that best reconstruct the phrase. Other methods generally use visual features that are generated from an input text query, and match them to image regions to find the object of interest [2,27].

A crucial step in these works is generating text captions from the images to describe objects and their relations. These captions are used to recognize the best match to the input query text. Methods based on recurrent neural networks [8,28,35] shown to be effective for caption generation. In the proposed method, we also generated image captions and evaluated them, but since this work is focused only on retrieving objects and their relations based on natural language, image captions are generated as a set of tuples including detected objects and their spatial relations. Moreover, our method is similar to these works, as we also compared the text query with the detected objects and their spatial relation, but different as in other works they trained neural networks, mostly based on long short-term memory (LSTM), to learn the similarity and alignments of them. However, in our method, the process is simplified using a similarity function which makes the need of training and required datasets unnecessary and also circumvents the non-transparency of approaches that solely use machine learning methods.

VQA is the task of answering a natural language question about images, and is sometimes referred as a “visual turing test” [10,25]. A large variety of VQA algorithms have been proposed in recent years, and in all methods features are extracted both from the image and a corresponding query text and then combined and feed to a classifier to predict an answer. Classical approaches are given in [21,31], which both used a semantic parser and instead of learning compositional operation they depend on fixed logical inferences. Recently proposed approaches differ in how they combine image and question features to infer an answer. In [1,18,42], features are merged by concatenation, element wise addition, and multiplication, and fed to a neural network or linear classifier to predict the answer. In [37], a CNN- and LSTM-based model is introduced that first recovers a structural scene representation from the images of various block world scenarios and then translates a natural language question into a program, which then is used to obtain an answer considering the given scene representation.

In [16,21], the authors use attention models to achieve better alignment between text and visual features. Similarly, the authors in [38] use language and visual attention mechanisms to map a natural language expression describing parts of an image, into the corresponding visual percepts.

In [34], the authors developed a method that learns to identify image regions that are most relevant to a given text query and uses these regions to answer the question. The work in [36] proposed a spatial inference method, spatial memory network for VQA, to answer questions about images. The method uses a two-hop model; in the first hop, the attention process extracts the image regions which correspond to individual words in the question, and in the second hop it predicts the answer using collected fine-grained evidence from the first hop and embeddings of the entire question. The authors in [24] introduce ViLBERT (short for Vision-and-Language BERT) which learns joint representations of vision and natural language features and visual language grounding is aimed to be a pretrainable and task agnostic.

The work on *relationship detection* is most similar to our work in as much that the natural language processing component is kept simple and focus is on the detection of relationships

between two objects in an image. In [7], the authors introduce an approach to detect visual relationships between two objects in an image using deep relational networks (DR-Net). Similar to our approach, bounding boxes of two objects in an image are extracted and an output of the triplet form (s, r, o) is inferred, where r describes the relation (e.g., spatial relation) between two objects s and o . The approach in [7] exploits spatial configurations and statistical evidence among the two objects and their relation via a deep relational network. In the paper [39], the authors detect, utilizing various versions of CNN, undetermined relationships between objects, that is, relationships that are not labeled as such or have false labels (e.g., a guitar being labeled as a lamp). The authors use a similarity measure and frequencies of the two objects and their relationship to transform these into probability distributions. In [40], the model is trained to detect triplets of the form (s, r, o) combining a semantic inference module and visual features. After a feature fusion step, the relationship between the two objects in question is predicted. The authors in [43] utilize Faster R-CNN ([32]) for object detection, followed by a relationship prediction composed of feature-level and label-level prediction for learning the objects-relationship triplets. In [22], a reinforcement learning framework for detecting visual relationships is proposed. The approach utilizes a directed semantic action graph for a given image for the prediction of the objects-relationship triplets and takes into account global context cues that describe the interactions among different objects in the image.

In all of these works, the attention is on extracting features from the query text and image and developing neural networks, mostly deep CNNs, to learn embeddings of features such that the best answer to the question is selected. In our approach, instead of utilizing CNNs, we used a function for each type of question based on the probability of detected objects, generated captions of the image and semantic similarity analysis of the query text. In this way, the developed system does not require training and the process of answer selection is less computational demanding and complex in comparison to deep neural networks as less number of parameters are introduced. It enables us to identify the exact shortcomings in the language grounding process and improve its performance. Furthermore, since we do not rely on training data in the process of embedding text and image features for answer selection, the proposed approach has advantages for under-resourced languages in which training data either does not exist or it is difficult to obtain, in comparison with existent datasets for other languages such as English.

Undoubtedly, there are many advantages with purely machine learning based approaches; however, it also has become clear that unwanted bias or discrimination is perpetuated due to already biased data or the machine learning approach (see [12] for an overview). In particular, the authors in [41] describe the bias found in visual language grounding. There are many ways to mitigate unwanted bias or discrimination in the learned model (for example, [5,17]), and one approach is to build hybrid systems that utilize the advantages of machine learning methods and algorithmic approaches, resulting in more transparent and interpretable systems.

3 Overview

Our system has three parts for processing the input image and the text query input, namely *image analysis*, *spatial relations analysis* and *text query analysis*. The image analysis and spatial relations analysis are independent processes from the text query analysis. The input to the image analysis is an image, and the output is a set of bounding boxes along with object categories (object labels) linguistically describing detected objects in the image (e.g.,

“woman,” “car”). The bounding boxes and object categories are used as input to the spatial relations analysis, which outputs spatial relation words (e.g., “left,” “under,” “behind”) for pairs of bounding boxes. The input to the text query analysis is a text query, and the output is n -tuples of linguistic symbols (e.g., noun) describing objects and spatial relations relevant for the processing of the query.

The output of the three parts is used as input for the algorithms for *language grounding*, where the n -tuples are mapped to the generated bounding boxes, object categories, and spatial relations. The general approach is to combine probabilistic measures of object categories and spatial relations generated from the image with measures of word similarity, such that the most probable grounding can be done.

In the following sections, image analysis, spatial relations analysis, text query analysis, and the algorithms for language grounding are described in more detail.

4 Image analysis

Given an image, the purpose of the image analysis part is to generate bounding boxes for identified objects in the image along with associated probabilities for the identified objects being of a certain object category (such as *person*, *monitor*, *car*). We used YOLO V2 [30], a pretrained neural network that recognizes $N_{OBJ} = 80$ object categories. It is very fast, which makes it a good candidate for real-time systems (for more details see reference [30]). We let $O = \{O_1, O_1, \dots, O_{N_{OBJ}}\}$ be the set of object category labels. Given an image I , YOLO returns a set of bounding boxes $B = \{b_1, b_2, \dots, b_{N_{BB}}\}$, $N_{BB} \geq 2$, where each bounding box b_i has associated values $op_{i,k}$ estimating the probability that b_i contains an object of category k , $1 \leq k \leq N_{OBJ}$. Each bounding box b_i also has associated image coordinates $[x_{1i}, y_{1i}, x_{2i}, y_{2i}]$, where (x_1, y_1) and (x_2, y_2) represent upper left and lower right corners, respectively. In our algorithms, we consider for each bounding box b_i the object categories with the five highest probabilities $op_{i,k}$, and set the other $op_{i,k}$ to zero.

5 Spatial relations analysis

We want the system to identify and correctly denote the spatial relation between pairs of bounding boxes in an image. A classifier was constructed to map the coordinates of pairs of boxes to an integer $k \in \{1, 2, 3, 4, 5, 6\}$, corresponding to an element in $SR = \{SR_1, SR_2, \dots, SR_{N_{SR}}\}$, with spatial relations: “left,” “right,” “top,” “under,” “front,” and “behind,” respectively. We used a weighted average probabilities network (WAP), with inputs from three classifiers with the same weights: a multilayer perceptron (MLP) with 4 layers and 10 neurons in each, a K -nearest neighbors with $K = 5$, and a support vector machine (SVM). The classifiers were trained using 505 images with a total of 1515 manually labeled spatial relations between bounding boxes. The bounding boxes were defined as $b_i = [x_{1i}, y_{1i}, x_{2i}, y_{2i}]$ and $b_j = [x_{1j}, y_{1j}, x_{2j}, y_{2j}]$. The values were scaled to make them size and location invariant:

$$S_1 = x_{2i} - x_{1i}, S_x = \frac{(x_{2j} - x_{1j})}{S_1}, L_x = \frac{(x_{1j} - x_{1i})}{S_1}, \quad (1)$$

$$S_2 = y_{2i} - y_{1i}, S_y = \frac{(y_{2j} - y_{1j})}{S_2}, L_y = \frac{(y_{1j} - y_{1i})}{S_2}, \quad (2)$$

$$Z = (S_x, S_y, L_x, L_y). \quad (3)$$

The vector Z was used as input to the classifiers. Data were split into training and testing sets, and the WAP classifier achieved 72.7% accuracy in fivefold cross-validation.

Since the WAP classifier outputs probability estimates for all 6 spatial relations $k \in \{1, 2, 3, 4, 5, 6\}$, these values were used as estimates of $sp_{i,j,k}$ (the probability that the spatial relation between the bounding boxes b_i and b_j is k).

Summarizing, for a given image I , we have a set of bounding boxes, $B = \{b_1, b_2, \dots, b_{N_{BB}}\}$, and for each bounding box b_i we have the five most probable object categories $O_k \in \{O_1, O_2, \dots, O_{N_{OBJ}}\}$ associated with it. In addition, we have for all pairs (b_i, b_j) , $i \neq j$ the probability $sp_{i,j,k}$ of the pair having a spatial relation k , where $k \in \{1, 2, 3, 4, 5, 6\}$ (corresponding to the spatial relations {"left," "right," "top," "under," "front," "behind"}, respectively). For example, $sp_{1,2,3}$ denotes the probability that object 1 is on top of object 2. In order to achieve a grounding between text and image, we later compute the similarity between the text labels obtained from the image analysis, and the text labels obtained from the query analysis.

6 Analysis of queries

We consider three types of queries:

1. *Attention queries* (type 1). Attention queries are commands in imperative form, beginning with a verb, containing two nouns and a spatial relation between the nouns. For example, the attention query "find the person to the right of the monitor" begins with the verb "find," has the two nouns "person" and "monitor," and a spatial relation "to the right." Our system returns two bounding boxes containing objects that correspond to the two nouns in the query and that have the spatial relation described in the query.
2. *Relation queries* (type 2). Relation queries begin with the word *where* and contain a noun. For example, the relation query "where is the person?", contains the noun "person." Our system returns a bounding box containing the noun in the query, and another box that contains another object in the image which is not linguistically expressed in the query. Our system returns two bounding boxes, one containing an object equal or similar to the noun in the query (e.g., "person" or "human"), another bounding box, and the spatial relation between the two bounding boxes (e.g., "to the right of"). The name of the object in the latter box is also returned.
3. *Identification queries* (type 3). Identification queries begin with the word *what*, contain a spatial relation and a noun. For example, the identification query "what is to the right of the monitor?" contains the spatial relation "to the right of" and the noun "monitor." Our system returns two bounding boxes, one containing the noun in the identification query (e.g., "monitor"), and the other having the given spatial relation to the first box. The name of the object in the latter box is also returned.

Given a query q , we use the Stanford CoreNLP parser [26] to extract syntactic categories in q . The syntactic categories determine the type of the query (i.e., 1, 2, or 3 as described above). We analyze the input query on three levels, namely clausal, phrasal and on word level. Table 1 shows the extracted syntactic categories for each type of query in tabular form. In particular, the syntactic categories and their quantities on the word level determine the type of input query. For example, as shown in Table 1, a query q is of type 1 if q has a syntactic category VB and three nouns NN .

Table 1 The syntactic categories and their quantity that determine whether a text query is of Type 1 (attention query), Type 2 (relation query) or Type 3 (identification query)

Constituent (i.e., Level)	Syntactic cat. Type 1	Syntactic cat. Type 2	Syntactic cat. Type 3
Clause	<i>S</i>	<i>SBARQ</i>	<i>SBARQ</i>
Phrase	<i>VP, NP</i>	<i>WHADVP, NP</i>	<i>WHNP, NP</i>
Word	<i>VB, NN (3x)</i>	<i>WRB, NN</i>	<i>WP, NN (2x)</i>

The syntactic categories *S* and *SBARQ* indicate a declarative clause and a question initiated by a *wh*-word. The syntactic categories *VP, NP, WHADVP* and *WHNP* stand for verbal phrase, nominal phrase, wh-adverb phrase and wh-noun phrase, respectively. The syntactic categories *VB, NN, WRB* and *WP* indicate verbs, nouns, wh-adverbs and wh-pronoun, respectively

Once we have extracted the syntactic categories and determined the type of query, we create tuples that are later used for the language grounding. In particular, we consider the words that are of syntactic category *NN* in the order in which they appear in the query:

- Given a type 1 query, we consider the words that are of category *NN*, that is, NN_1, NN_2 and NN_3 . We let NN_1 be $entity_1$, NN_2 be sr_1 and NN_3 be $entity_2$ and create the tuple $\langle entity_1, sr_1, entity_2 \rangle$. For example, in the query “find the person to the right of the monitor” we have three words that are *NN*, namely “person,” “right” and “monitor” and we generate the tuple $\langle person, right, monitor \rangle$, describing that the person is to the right of the monitor.
- Given a type 2 query, we consider the single word of category *NN* and let NN be $entity_1$ and create $\langle entity_1 \rangle$. For example, the query “where is the person?” has one *NN*, namely “person” and we generate $\langle person \rangle$.
- Given a type 3 query, we consider the two words that are *NN*, that is NN_1 and NN_2 . We let NN_1 be sr_1 and NN_2 be $entity_2$ and create $\langle sr_1, entity_2 \rangle$. For example, in “what is to the right of the monitor?”, the words “right” and “monitor” are *NNs* and we generate the tuple $\langle right, monitor \rangle$.

7 Natural language grounding

After processing the input image and text query, the following data are available:

- N_{BB} bounding boxes and estimated probabilities $op_{i,k}$ for bounding box b_i containing an object of category k .
- Estimated probabilities $sp_{i,j,k}$ for bounding boxes i and j having a spatial relation k .
- Query type and tuples containing object names and spatial relation labels extracted from the query q .

These data are used in algorithms for generation of appropriate responses to the queries. To improve matching of object labels given in the query with labels returned from the image analysis, a method to identify semantic similarity between words was employed. We used the similarity function from spaCy¹ [14], an open-source library for natural language processing. Based on the angle between two input word vectors, it computes a similarity measure between 0 and 1. For example, the words “person” and “person” have a similarity value of 1.0, while “person” and “woman” have a value of 0.84, and “monitor” and “person” a value of 0.79. In

¹ www.spacy.io.

Fig. 1 Image with three detected bounding boxes. A list of possible objects in each box, and corresponding probabilities is given in Table 2

Table 2 Object labels and probabilities for the 5 most probable objects in each one of the 3 bounding boxes shown in Fig. 1

Bounding box 1	Bounding box 2	Bounding box 3
Person 0.81	Monitor 0.63	Chair 0.85
Chair 7×10^{-3}	Refrigerator 0.01	Toilet 2.3×10^{-4}
Oven 3.2×10^{-4}	Microwave 0.007	Sofa 6.63×10^{-6}
Dining table 2.1×10^{-4}	Laptop 0.005	Bench 4.79×10^{-6}
Book 1.9×10^{-4}	Oven 0.003	Suitcase 3.08×10^{-6}

the following description of algorithms, the similarity function is denoted *sim* and takes two strings as input arguments. Similarity is computed for object categories $O_1, O_2, \dots, O_{N_{OBJ}}$, and object names appearing in the queries (see Table 4). Similarity is also computed for spatial relations $SR_1, SR_2, \dots, SR_6$ and words expressing spatial relations in the queries (see the example in Table 5).

Antonyms of spatial relations were also considered to improve performance. For example, when trying to find a person to the right of a monitor, the system also considered finding a monitor to the left of a person. Antonyms were extracted using WordNet [29] alongside the NLTK module [4].

In the following subsections, the algorithms for each one of the three query types are described in detail.

7.1 Attention queries

Algorithm 1 Input: An image I and an attention query q (such as “find the person to the right of the monitor.”), containing an object denoted entity₁, a spatial relation denoted sr₁, and a second object denoted entity₂.

Output: A bounding box β_1 containing an object of type $entity_1$, and a bounding box β_2 , where sr_1 describes the spatial relation between β_1 and β_2 .

Method: Calculate β_1 and β_2 as follows:

1. Syntactically analyze q to generate $entity_1$, sr_1 , and $entity_2$, where sr_1 is a spatial relation word, $entity_1$ and $entity_2$ are nouns referring to objects (see Sect. 6).
2. Generate the inverse relation sr_2 as the antonym of sr_1 .
3. Analyze I to generate a set of N_{BB} bounding boxes: $B = \{b_1, b_2, \dots, b_{N_{BB}}\}$, where each b_i has associated image coordinates $[x_{1i}, y_{1i}, x_{2i}, y_{2i}]$, and values $op_{i,k}$, $1 \leq k \leq N_{OBJ}$, where $op_{i,k}$ is the probability that b_i contains an object of type k (see Sect. 4).
4. Compute probabilities $sp_{i,j,k}$ for the spatial relation between bounding boxes b_i and b_j being SR_k , for $1 \leq i, j \leq N_{BB}$, $i \neq j$, $1 \leq k \leq N_{SR}$ (see Sect. 5).
5. Compute

$$(i, j, k, m, n) = \underset{1 \leq i, j \leq N_{BB}, 1 \leq k \leq N_{SR}, 1 \leq m, n \leq N_{OBJ}}{\operatorname{argmax}} (op_{i,m} \times op_{j,n} \times \operatorname{sim}(O_m, entity_1) \times \operatorname{sim}(O_n, entity_2) \times z_{i,j,k}) \quad (4)$$

where

$$z_{i,j,k} = \max(sp_{i,j,k} \times \operatorname{sim}(SR_k, sr_1), sp_{j,i,k} \times \operatorname{sim}(SR_k, sr_2)). \quad (5)$$

6. Let $\beta_1 = b_i$ and $\beta_2 = b_j$.

The steps in the algorithm are illustrated in the following example, for the input query “find the person to the right of the monitor.”, and the input image shown in Fig. 1 (All images are from the Visual Genome dataset [20] and used with permission (see acknowledgments for details)):

1. The syntactic analysis of q yields: $entity_1 = \text{“person”}$, $sr_1 = \text{“right”}$, $entity_2 = \text{“monitor”}$.
2. The antonym of sr_1 is computed as $sr_2 = \text{“left”}$.
3. In the input image, three bounding boxes are generated. The five most probably objects for each box are listed in Table 2.
4. The neural network estimates probabilities $sp_{i,j,k}$ for the six spatial relations k for all pairs (i, j) of bounding boxes, as shown in Table 3. For example, box 1 is located to the right of box 2 with probability 0.919, which is the value assigned to $sp_{2,1,2}$.
5. The most likely bounding boxes i and j are computed by solving the maximization problem in Eqs. 4 and 5. The optima is achieved for $i = 1$ and $j = 2$.
6. Let $\beta_1 = b_2$ and $\beta_2 = b_1$.

7.2 Relation queries

Algorithm 2 Input: An image I and a relation query q (such as “where is the person?”), containing an object denoted $entity_1$.

Output: An object name o , a bounding box β_1 containing $entity_1$, and a bounding box β_2 containing an object of category o , and sr describing the spatial relation between β_1 and β_2 .

Method: Calculate o , β_1 , β_2 , and sr as follows:

1. Syntactically analyze q to extract $entity_1$ (see Sect. 6).
2. Same as step 3 in Algorithm 1.

Table 3 Probabilities $sp_{i,j,k}$ for spatial relations k between bounding boxes i and j in Fig. 1

Bounding boxes (i, j)	Left	Right	Top	Under	Front	Behind
1,2	0.007	0.919	0.005	0.003	0.024	0.042
1,3	0.222	0.001	0.077	0.001	0.076	0.622
2,3	0.265	0.010	0.024	0.008	0.016	0.676

3. Same as step 4 in Algorithm 1.

4. Compute

$$(i, j, k, m, n) = \operatorname{argmax}_{1 \leq i, j \leq N_{BB}, 1 \leq k \leq N_{SR}, 1 \leq m, n \leq N_{OBJ}} (op_{i,m} \times op_{j,n} \times \operatorname{sim}(O_m, \text{entity}_1) \times sp_{i,j,k}) \quad (6)$$

5. Let $o = O_n$, $\beta_1 = b_i$, $\beta_2 = b_j$, and $sr = SR_k$.

The steps in the algorithm are illustrated in the following example, for an input query “where is the person?”, and the input image shown in Fig. 1:

1. entity_1 is extracted from q as “person.”
2. Same as step 3 in Algorithm 1.
3. Same as step 4 in Algorithm 1.
4. The most likely object, bounding boxes, and spatial relation are computed by solving the maximization problem in Eq. 6. The optima is achieved for $n = 1$ (corresponding to the first object category in Table 2), $i = 1$, $j = 2$ and $k = 2$ (spatial relation “right”).
5. Let $o = O_n$, $\beta_1 = b_i$, $\beta_2 = b_j$, and $sr = SR_k$.

7.3 Identification queries

Algorithm 3 Input : An image I and an identification query q (such as “what is to the right of the monitor”), containing a spatial relation denoted sr_1 and an object denoted entity_2 .

Output : An object category o , a bounding box β_1 containing an object of category o , and a bounding box β_2 containing entity_2 .

Method: Calculate β_1 , β_2 , and o as follows:

1. Syntactically analyze q to extract an object entity_2 and a spatial relation sr_1 (see Sect. 6).
2. Generate the inverse relation sr_2 as the antonym of sr_1 .
3. Same as step 3 in Algorithm 1.
4. Same as step 4 in Algorithm 1.
5. Compute

$$(i, j, k, m, n) = \operatorname{argmax}_{1 \leq i, j \leq N_{BB}, 1 \leq k \leq N_{SR}, 1 \leq m, n \leq N_{OBJ}} (op_{i,m} \times op_{j,n} \times \operatorname{sim}(O_n, \text{entity}_2) \times z_{i,j,k}), \quad (7)$$

where

$$z_{i,j,k} = \max(sp_{i,j,k} \times \operatorname{sim}(SR_k, sr_1), sp_{j,i,k} \times \operatorname{sim}(SR_k, sr_2)). \quad (8)$$

6. Let $\beta_1 = b_i$, $\beta_2 = b_j$, and $o = O_m$.

The steps in the algorithm are illustrated in the following example, for an input query “what is to the right of the monitor.”, and the input image shown in Fig. 1:

Table 4 Similarity value between labels of detected objects from image in Fig. 1 and objects in the query

Query objects	Monitor	Refrigerator	Person	Chair	Chair	Toilet
Person	0.79	0.77	1	0.78	0.78	0.67
Monitor	1	0.75	0.79	0.75	0.75	0.66

Table 5 Examples of similarity values of spatial relations

Query relations	Left	Right	Top	Under	Front	Behind	Above	Below	Next	On
Right	0.44	1	0.66	0.51	0.64	0.51	0.57	0.43	0.66	0.41
Left	1	0.44	0.52	0.22	0.33	0.39	0.47	0.34	0.45	0.30

1. Entities and a spatial relation are extracted from q as: $entity_2 = \text{"monitor,"}$ $sr_1 = \text{"right."}$
2. The antonym of sr_1 is computed as $sr_2 = \text{"left"}$
3. Same as step 3 in Algorithm 1
4. Same as step 4 in Algorithm 1
5. The most likely bounding boxes and object are computed by solving the maximization problem in Eqs. 7 and 8. The optima is achieved for $m = 1, i = 1, j = 2$ and $k = 2$.
6. Let $o = O_m, \beta_1 = b_i, \beta_2 = b_j$.

8 Evaluation

The performance of the developed system was assessed by test users who reported how well the system's generated answers matched their own view for a given set of images and questions. The users used a test program showing a sequence of images, see Fig. 2. At first, the image was shown with detected bounding boxes. The user was then asked to compose a question fitting a given template corresponding to one of the three query types. For type 1, the template was "Find the $\langle object1 \rangle < spatialrelation \rangle \langle object2 \rangle$," for type 2 "Where is the $\langle object \rangle$?", and for type 3 "What is $\langle spatial relation \rangle \langle object \rangle$?". After completing the query, the system produced and displayed an answer in the field "Systems Output Text," and the user clicked on one of the fields "Answer correct" "Answer Not Correct," or "Not Sure." For the second option, the reason could be specified as either "Wrong Spatial Relation," "Wrong Object Detection," or both.

The system also produced a number of image captions (depending on the number of detected objects) describing the relation between objects in the image. The user assessed these captions by entering the number of accepted captions. Each user assessed the system with 12 images, six of which were common to all users. These six images were used to analyze how differently people interpret spatial relations in a scene.

Thirty users were recruited for the evaluation. They were all university students with at least a Masters degree, and all spoke good English. The users analyzed 186 different images, generated 1080 queries, used 75 different object names, and 10 different words describing spatial relations to form queries. The spatial relations include: behind, front, right, left, above, top, on, under, below and bottom. A total of 2005 image captions were generated and assessed.

Fig. 2 The test program used for evaluation of the system. At first an image with unlabeled bounding boxes is displayed. A question is input by the user and the system displays the system output. The user then assesses whether this output is correct or not. Image captions are also generated for more extensive analysis

Table 6 System performance for different number of detected object labels ($op_{i,k}$) with and without the semantic similarity function

Number of labels	With similarity function			Without similarity function		
	Correct	Not correct	Accuracy (%)	Correct	Not correct	Accuracy (%)
5	885	193	81.9	656	422	60.7
4	885	193	81.9	656	422	60.7
3	885	193	81.9	656	422	60.7
2	882	196	81.6	653	425	60.4
1	862	216	79.8	644	434	59.6

9 Results and analysis

According on the test users' responses, the system correctly answered 81.9% of the 1080 posed queries in the evaluation. Of the 18.1% incorrect answers, 62.2% were caused by incorrect detection of spatial relations, and 37.8% by incorrect object classification. 68.9% of the generated captions were assessed as correct.

One strength with the presented solution is that several (in the presented results five) object labels were simultaneously considered for each detected bounding box. To investigate the value of this approach, performance was computed for a system considering also 1–4 object labels. As shown in Table 6 (left part), performance dropped from 81.9 to 79.8% when considering only one object label. The reason is that the target object in the query sometimes does not match the label with the highest probability, even if the bounding box really contains the target object. In such cases, an incorrect bounding box may be selected if only the object with highest probability is considered. This situation is exemplified in Fig. 4.

The used YOLO system can detect 80 objects of different classes. Without additional precautions, our system would not be able to handle queries related to other nouns. As an example, the query like “find the “woman” cannot be correctly processed, since “woman” is not one of these 80 labels. We overcame this limitation by computing similarity values

Table 7 Object labels and their probabilities for the bounding boxes in Fig. 4

Red	Blue	Yellow	Green
Person 0.79	Cup 0.72	Bottle 0.80	Bottle 0.61
Chair 3.3×10^{-2}	Bowl 7.5×10^{-2}	Refrigerator 4.1×10^{-2}	Cup 0.033×10^{-2}
Teddy bear 1×10^{-3}	Spoon 2.3×10^{-3}	Cup 0.015	Book 1×10^{-3}

between object categories in the query and the 80 categories that can be detected in images. In this way, we can handle a larger vocabulary, as demonstrated in Fig. 3. In the shown example, the system detected the “chair,” “monitor,” and a “person.” For the query “Where is the woman?” the similarity between “woman” and the detected “person” is computed as 0.84, which is sufficiently high for the correct bounding box to be selected by Algorithm 2. The system creates a correct response by labeling the bounding box with “woman,” and generating the text “woman right of monitor.”

The value of the similarity function increases when considering more than one object label. Figure 4 shows four objects, a person, a bowl, a milk box and a cereal box, in red, blue, yellow and green bounding boxes, respectively. YOLO assigns highest probabilities to the object labels “person,” “cup” (the bowl), “bottle” (the milk box) and “bottle” (the cereal box) as shown in Table 7. For the query “Where is the bowl?,” “bowl” does not match any of these objects. Additionally, since the similarity value between “person” and “bowl” (0.78) is higher than between “bowl” and “cup” (0.73), and between “bowl” and “bottle” (0.76), “person,” would be selected in the system’s response. On the other hand, if we used more than one object label for each bounding box, “bowl” would be the second most probable object label for the blue bounding box. As a result, the blue bounding box would be returned as the system response.

To investigate the value of the similarity function, we computed performance also with this function disabled, thus affecting matching of both object names and names for spatial relations in the algorithms. As can be seen in Table 6 (right part), discarding the similarity function reduced the system performance by about 20%. In the example illustrated in Fig. 6, the similarity function was disabled with the results that the object name “armchair” given in the query did not match any of the labels of the objects detected in the image. The system then generated the correct, but irrelevant, answer “person right of monitor.” When using the similarity function, “armchair” was matched with the detected object “chair,” and the image caption “person behind armchair” was generated. This shows the importance of the similarity function in the proposed solution.

Analysis of the data from test users’ responses gave us new insights in how spatial relations in images are perceived and described. We identify two kinds of a scene interpretations and denote them *global* and *local*. When a spatial relation between objects in an image is based on the perspective of the camera, it is defined as global. When it is based on the perspective of an object or person in the image, it is defined as local. Figure 5 clarifies the concepts. In this figure, the description “monitor to the left of the woman” is an example of a global relation, while “monitor in front of the woman,” is an example of a local relation. The neural network for classification of spatial relations was trained and tested with data labeled with global relations. Hence, the trained network did not model cases where users interpret relations locally. The relatively low accuracy (68.9%) for the generated captions may be caused by this and indicates that people tend to interpret and express spatial relations locally to a significant extent. Nevertheless, the system often managed to infer the correct bounding

Fig. 3 The designed system is not limited to the number of class labels the object detection system accepts. In the shown example, the word “woman” appears in the query, but is not recognized by Yolo. However, by computing linguistic similarity between “woman” and “person,” which is the object label detected by Yolo, the correct bounding box is selected and labeled “woman” (color figure online)

Fig. 4 Example showing the value of considering more than one object type for each bounding box. The image analysis part outputs bounding boxes with assigned probabilities for object classes (see Table 7). The objects with highest probability in the red, blue, yellow and green bounding boxes are *person*, *cup*, *bottle* and *bottle*, respectively. Using only the most probable object, the query “where is the bowl?,” results in an output highlighting the red bounding box. If the two most probable objects in each bounding box are considered, the system correctly highlights the blue bounding box (color figure online)

boxes and object names, even if the most likely spatial relation predicted by the neural network did not match the test user’s assessment. This robustness is due to the probabilistic approach through which several spatial relations are predicted and considered, and a low probability $sp_{i,j,k}$ for a spatial relation may be counter weighted by high probabilities $op_{i,m}$, $op_{j,n}$, for the detected object classes (see Eqs. 4 and 5) (Fig. 6).

Fig. 5 An example of global (a) and local (b) ways of describing spatial relations. Investigating queries and answers by test users, we found that some address the monitor as “to the left of the woman” (globally), and some as “in front of the woman” (locally)

Fig. 6 With the similarity function disabled, the query object “armchair” is not grounded to the chair in the image, and the system gives the incorrect output “person right of monitor” (color figure online)

10 Comparison to other work

There are purely machine learning-based systems to visual language grounding and quite a few approaches that can be characterized as hybrid approaches combining some kind of structural analysis with machine learning (for example, [21,22,37]), which our approach falls into. As outlined in Sect. 2, one of the unwanted outcomes in purely based machine learning approaches is the unwanted bias and debiasing strategies may operate directly on the data or on the language model, whereas other approaches try to find evidence what, for example, vector representations actually encode (as, for example, in [6,13]). One bias-aware approach is to build hybrid systems that incorporate some structural method or algorithmic approach. In the following, we compare our work in more detail to several approaches that are most similar to our approach in so far as these approaches learn triplets of the form (o_1, r, o_2) , that is the relation r between two objects o . It is not possible to make a purely quantitative comparison since all of these approaches use recall as an evaluation metric for performance measure, whereas we use accuracy. In addition, we evaluate our performance on test users.

In [7], the authors introduce deep relational networks (DR-Net) and given the bounding boxes of two objects, a triplet (s, r, o) is predicted, where r describes the relation between the two objects s and o . The approach exploits spatial configurations and statistical evidence among the two objects and their relation via a deep relational network. The authors test their approach on two datasets VRD [23] and sVG which is a larger dataset constructed from the Visual Genome dataset [20]. The training dataset in [7] is considerably larger than ours (with 108K images and 998K relationship instances). In addition, they operate solely over object labels and do not process text queries for visual grounding. The Recall values on the dataset sVG are 88.26 (Recall@50) and 91.26 (Recall@100). The accuracy for our approach is at 81.9, and the assumed lower performance (taking into account some precision measure) may be due to the fact that we do not use the most probable class label if it does not describe the object in the image but consider the five most probable class labels, taking into account only class labels that actually describe the object in the image. For example, a child and a woman in illustrated images in [7] are both predicted as “man,” whereas in our approach they would not or at least not without affecting the accuracy. This is an example how (gender) biased pure machine learning methods can be by, for example, not discriminating between the class label *man* and the actual person in the image.

In [39], the authors detect so-called undetermined relationships between objects, that is, relationships that are not labeled or have false labels (e.g., a *guitar* hanging on the wall being labeled as a *lamp*). The authors use a similarity measure (e.g., word2vec) and frequencies of all relation triplets in the training set to get a probability distribution. These frequencies might again amplify unwanted bias, or reflect a biased dataset (that does not contain images with electric guitars hanging on the wall, which probably is a very common way to store and display guitars). Visual Genome is the dataset used for their performance evaluation. Their Recall is 14.4 (Recall@50) and 16.5 (for Recall@100) for relationship prediction. In our approach, the probability of a triplet such as (guitar, on, wall) would likely have low probability values as well, however would hopefully be more robust due to the algorithmic nature of grounding visual percepts into the image.

The authors in [40] use a semantic inference module based on word vector representations, attention mechanism of the global image context, and feature fusion to learn relation triplets (object1, predicate, object2). Their model is also validated on the Visual Genome Relationship dataset with per type predicate classification accuracy with Recall rates 65.0 (Recall@50) and 67.1 (Recall@100).

In [22], the authors introduce a reinforcement learning framework for detecting relationships between objects and their attributes. Their approach systematically detects relationship and attribute instances according to a traversal scheme on a built directed semantic action graph for the image. In addition, to local analysis the authors incorporate also global analysis of the image for learning purposes. Their results for relationship detection on Visual Genome are 13.34 (Recall@50) and 12.57 (Recall@100).

11 Conclusion and future work

We presented a system for responding to three types of questions regarding objects and their spatial relations in given images. The answers comprised identification of objects in the image and generation of appropriate text. 81.9% of the generated answers were assessed as correct by 30 test users. The system’s robustness was demonstrated by the fact that it often correctly answered queries based on a local view on spatial relations, while it was trained on data with

a global view (see Sect. 9). This was an effect of the probabilistic approach that combine probabilities for object classes and spatial relations.

By using the semantic similarity function, our model overcame the problems with a limited number of object classes in pretrained network models. Flexibility regarding the varying ways users express spatial relations was also improved. Without the semantic similarity function, accuracy was reduced to 60.7%. Another feature that contributed to the high performance was that several object types were considered for each detected bounding box. The approach in which probabilities for object classes and spatial relations were combined with a measure of semantic similarity contributed to robustness as well as high performance.

In the proposed approach, we used a set of functions instead of training CNNs for measuring the similarity between text query and detected objects and also selecting the most probable answer. It resulted in a less complex and computationally demanding process which also discarded the need of training data. Therefore, the developed method can also be used for under-resourced languages, with minimal number of changes, in which the required training data are not available. The system could be further enhanced by training the algorithms with the user responses from the evaluation.

The automatic identification of spatial relations would benefit from a depth perspective, since humans easily perceive depth in images, and also denote spatial relations based on that, for example with “behind” and “front” relations. Methods for generation of depth in regular 2D images [3,9] could be investigated, and usage of 3D cameras would obviously be a potential approach.

Such an extension, along with incorporating more identifiable objects and relations between objects, could be relevant for Urban Search and Rescue Robots (USAR), where robots and human operators work together to locate humans after natural disasters such as flooding or earthquakes. For USAR situations, where, for example, the robot is in an environment unreachable to the human operator and information about the environment has to be exchanged via the robot’s remote cameras, language grounding algorithms would have to take into account the complexity of changing environments as well as perspectives of the robot.

Acknowledgements The authors gratefully acknowledge the permission to use the images from the Visual Genome dataset [20] under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>). The following changes were made to the original images: Fig. 2, three bounding boxes and indices added; Figs. 3, 4 and 6, blue and red bounding boxes and labels “person” and “monitor” added; Fig. 5, four bounding boxes added; Fig. 6, bounding boxes, arrows and green lines added. In addition, many thanks are due to the anonymous reviewers for their suggestions.

Funding Open access funding provided by Umeå University.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433 (2015)
2. Arandjelovic, R., Zisserman, A.: Multiple queries for large scale specific object retrieval. In: *BMVC*, pp. 1–11 (2012)
3. Battiato, S., Curti, S., La Cascia, M., Tortora, M., Scordato, E.: Depth map generation by image classification. In: *Three-Dimensional Image Capture and Applications VI*, vol. 5302, pp. 95–104. International Society for Optics and Photonics (2004)
4. Bird, S., Klein, E., Loper, E.: *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media Inc, Newton (2009)
5. Bolukbasi, T., Chang, K., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, December 5–10, 2016, Barcelona, Spain, pp. 4349–4357 (2016)
6. Dahlgren Lindström, A., Bensch, S., Björklund, J., Drewes, F.: Probing multimodal embeddings for linguistic properties: the visual-semantic case. In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 730–744. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020). <https://doi.org/10.18653/v1/2020.coling-main.64>
7. Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3298–3308 (2017). <https://doi.org/10.1109/CVPR.2017.352>
8. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634 (2015)
9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: *Advances in Neural Information Processing Systems*, pp. 2366–2374 (2014)
10. Geman, D., Geman, S., Hallonquist, N., Younes, L.: Visual Turing test for computer vision systems. *Proc. Natl. Acad. Sci.* **112**(12), 3618–3623 (2015)
11. Guadarrama, S., Rodner, E., Saenko, K., Zhang, N., Farrell, R., Donahue, J., Darrell, T.: Open-vocabulary object retrieval. *Robot. Sci. Syst.* **2**, 6 (2014)
12. Hellström, T., Dignum, V., Bensch, S.: Bias in machine learning—what is it good for? In: Saffiotti, A., Serafini, L., Lukowicz, P. (eds.) *Proceedings of the First International Workshop on New Foundations for Human-Centered AI (NeHuAI) co-located with 24th European Conference on Artificial Intelligence (ECAI 2020)*, Santiago de Compostella, Spain, September 4, 2020, *CEUR Workshop Proceedings*, vol. 2659, pp. 3–10. CEUR-WS.org (2020)
13. Hewitt, J., Liang, P.: Designing and interpreting probes with control tasks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, Hong Kong, China (2019). <https://doi.org/10.18653/v1/D19-1275>
14. Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017). To appear
15. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4555–4564 (2016)
16. Ilievski, I., Yan, S., Feng, J.: A focused dynamic attention model for visual question answering (2016). arXiv preprint [arXiv:1604.01485](https://arxiv.org/abs/1604.01485)
17. Islam, A.C., Bryson, J.J., Narayanan, A.: Semantics derived automatically from language corpora necessarily contain human biases (2016). *CoRR* **abs/1608.07187**
18. Kafle, K., Kanan, C.: Answer-type prediction for visual question answering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4976–4984 (2016)
19. Karpathy, A., Joulin, A., Fei-Fei, L.F.: Deep fragment embeddings for bidirectional image sentence mapping. In: *Advances in Neural Information Processing Systems*, pp. 1889–1897 (2014)
20. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* **123**(1), 32–73 (2017)
21. Krishnamurthy, J., Kollar, T.: Jointly learning to parse and perceive: connecting natural language to the physical world. *Trans. Assoc. Comput. Linguist.* **1**, 193–206 (2013)
22. Liang, X., Lee, L., Xing, E.: Deep variation-structured reinforcement learning for visual relationship and attribute detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017)

23. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision—ECCV 2016*, pp. 852–869. Springer International Publishing, Cham (2016)
24. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett R. (eds.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada*, pp. 13–23 (2019)
25. Malinowski, M., Fritz, M.: A multi-world approach to question answering about real-world scenes based on uncertain input. In: *Advances in Neural Information Processing Systems*, pp. 1682–1690 (2014)
26. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60 (2014). <http://www.aclweb.org/anthology/P/P14/P14-5010>. Accessed 1 May 2019
27. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–20 (2016)
28. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-rnn) (2014). arXiv preprint [arXiv:1412.6632](https://arxiv.org/abs/1412.6632)
29. Miller, G.A.: Wordnet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995). <https://doi.org/10.1145/219717.219748>
30. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger (2016). arXiv preprint [arXiv:1612.08242](https://arxiv.org/abs/1612.08242)
31. Ren, M., Kiros, R., Zemel, R.: Image question answering: a visual semantic embedding model and a new dataset. *Proc. Adv. Neural Inf. Process. Syst.* **1**(2), 5 (2015)
32. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)
33. Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction. In: *European Conference on Computer Vision*, pp. 817–834. Springer (2016)
34. Shih, K.J., Singh, S., Hoiem, D.: Where to look: focus regions for visual question answering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4613–4621 (2016)
35. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164 (2015)
36. Xu, H., Saenko, K.: Ask, attend and answer: exploring question-guided spatial attention for visual question answering. In: *European Conference on Computer Vision*, pp. 451–466. Springer (2016)
37. Yi, K., Wu, J., Gan, C., Torralla, A., Kohli, P., Tenenbaum, J.: Neural-symbolic VQA: disentangling reasoning from vision and language understanding. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett R. (eds.) *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada*, pp. 1039–1050 (2018)
38. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: MATTNet: Modular attention network for referring expression comprehension. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, pp. 1307–1315. IEEE Computer Society (2018)
39. Zhan, Y., Yu, J., Yu, T., Tao, D.: On exploring undetermined relationships for visual relationship detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5123–5132 (2019). <https://doi.org/10.1109/CVPR.2019.00527>
40. Zhang, L., Zhang, S., Shen, P., Zhu, G., Afaq Ali Shah, S., Bennamoun, M.: Relationship detection based on object semantic inference and attention mechanisms. In: *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR'19*, p. 68–72. Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3323873.3325025>
41. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., Chang, K.: Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In: Palmer, M., Hwa, R., Riedel, S. (eds.) *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017*, pp. 2979–2989. Association for Computational Linguistics (2017)
42. Zhou, B., Tian, Y., Sukhbaatar, S., Szlam, A., Fergus, R.: Simple baseline for visual question answering (2015). arXiv preprint [arXiv:1512.02167](https://arxiv.org/abs/1512.02167)
43. Zhu, Y., Jiang, S.: Deep structured learning for visual relationship detection. In: *AAAI* (2018)