

Not Everybody's Special: Using Neighbors in Referring Expressions with Uncertain Attributes

Amir Sadovnik
as2373@cornell.edu

Andrew Gallagher
acg226@cornell.edu

Tsuhan Chen
tsuhan@ece.cornell.edu

School of Electrical and Computer Engineering, Cornell University

Abstract

Referring expression generation is widely considered a basic building block of any natural language generation system. Generating these phrases, which can point out a single object from a group of objects, has been studied extensively in that community. However, to build systems which can discuss images in an intelligent way, it is necessary to consider additional factors unique to the visual domain. In this paper we consider the use of neighbors as anchors to create a referring expression for a person in a group image. We describe a target person using the people around him, when we cannot find a reliable set of attributes to describe the target himself. We first present a method for including neighbors in a referring expression, and discuss several ways of presenting this data to a user. We show through experiments that using descriptions with neighbors can significantly improve the probability of conveying the correct information to a user.

1. Introduction

Imagine you are at a party with many people, and need to point out one of them to a friend. Since it is impolite to point (and also since it is hard to follow the exact pointing direction in a big group), you decide to describe the target person to your friend in words. Most people would have an easy time deciding what information to include and build what is known in the Natural Language Processing field as a *referring expression*. For example, in Fig. 1, we might say: “The lady with the black hair” to describe person (a).

The task of generating these expressions requires a balance between two properties as stated by Grice’s Maxim of Quantity [12]. The maxim of quantity states:

1. Make your contribution as informative as is required.
2. Do not make your contribution more informative than is required.

In our context, in which the computer attempts to refer to

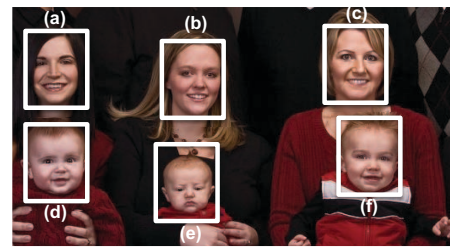


Figure 1. In this paper we show that using neighbor faces to create referring expressions can improve their accuracy. For example, trying to create a referring expression for face (d) using only his facial attributes might be difficult because he is very similar to face (f). However, our algorithm produces the expression, “Please choose the person to the left of the person who is a baby and is frowning”, which completely resolves this type of ambiguity.

a single person, we interpret these as follows. First, the description ideally refers to only the single target person in the group such that the listener (guesser) can identify that person. Second, the describer must try to make the description as short as possible.

However, there are cases when describing a target is difficult since it might be too similar to other people in the image. For example, when trying to refer to face (d) in Fig. 1 we might find it difficult since our vocabulary might not include attributes which differentiate faces (d) and (f). However, since face (e) is easily referred to as “frowning”, we can use him as an anchor and create the expression, “Please choose the person to the left of the person who is a baby and is frowning”, which identifies face (d) as required.

To determine when it is necessary to use neighbors we must be able to calculate the probability of a user guessing the correct face given an image and a description. We use our previous method from [22] to calculate this probability using attribute classifier scores in an efficient way.

This task represents an important part of a broader set of problems which address generating general descriptions for images. This is evident from the fact that referring expression generation is considered one of the basic building blocks for any natural language generation system [19].

When giving a general description one might be required to refer to specific objects within the scene, and since our attribute list might not contain ones with which the target varies from the distractors, using neighbor objects might be necessary. For example in Fig. 1, it is more helpful to say, “The child to the left of the frowning baby is Benjamin” rather than “The smiling child is Benjamin”. This type of referral is crucial in generating informative image captions. Our algorithm provides a method for selecting which neighbor should be selected and which attributes should be mentioned in such a case.

Another application involves navigation systems. Using a front-facing camera on a car and a GPS system, we can develop a system which provides more intuitive driving directions. Instead of saying: “Turn right in 200 feet” it might be more useful to say: “Turn right at the stop light to the right of the tree” or “Follow the green car to the left of the red building”. Since street scenes contain many objects which are similar to each other such as traffic signs, traffic lights and buildings, the option of using other objects as anchors might be crucial to creating efficient referring expressions. Although in this paper we present the algorithm for people description, it is not confined to this specific object. By utilizing object detection algorithms in addition to other attribute classifiers, a system like this can be realized.

Our main contributions are: We present a method for describing people using a neighboring anchor person when the target face cannot be described with a high confidence. We show that although this complicates the description by adding an additional spatial term, it significantly improves the probability of guessing correctly vs. using only the target’s description. We also show that the way the description is presented to the user is crucial for success.

1.1. Previous Work

There has been active research on referring expression generation in the NLG community for 20 years. Most do not consider anchor objects, and begin with a setup in which there exists a finite object domain D , each with attributes A . The goal is to find a subset of attribute-value pairs which is true for the target but false for all other objects in D . Different datasets have been used to evaluate such descriptions such as the TUNA database [10]. We build on this work from a computer vision point-of-view, using actual attribute predictions and spatial information.

One of the earliest works include Dale’s *Full Brevity* algorithm [3] which finds the shortest solution by exhaustive search. Since this results in an exponential-time algorithm two main extensions were introduced in [4]. The Greedy Heuristic method chooses items iteratively by selecting the attribute which removes the most distractors that have not been ruled out previously until all distractors have been ruled out. The Incremental Algorithm considers an addi-

tional ranking based on some internal preference of what a human describer would prefer, in an effort to produce more natural sounding sentences. Our goal is the same (to produce discriminative descriptions), but we consider the confidence scores of actual attribute classifiers, and add additional spatial relationships.

Other extensions to these three main algorithms have been proposed. For example, Horacek proposes an algorithm which deals with conditions of uncertainty [14]. This method is similar to the one presented in [22] since it does not rely on the fact that the describer and the listener agree on all attributes. However, our algorithm differs in important ways. First, we provide a method for efficient calculation under uncertain conditions whereas in Horacek’s paper the calculation is computationally expensive. In addition, Horacek’s definition of the uncertainty causes is heuristic, but we use calculated uncertainties of classifiers. And, in contrast to [14], we provide experimental data to show our algorithm’s strength.

In the language domain there has also been previous work regarding the inclusion of relational properties. Many rely on the assumption that using relationships between objects is less preferential than using attributes of the target object itself, even though this has been shown to not always be correct (See [24]). Golland et al. [11] uses a game theoretic approach to select which objects to use as anchors. By collecting human annotations and using spatial features they learn which anchors will provide the greatest utility. However, they do not use any attributes for the objects mentioned. Our use of neighbors is similar to the work done by Kelleher et al. [15] in that we first attempt to describe the object by itself, and only use referents if necessary.

Krahmer *et al.* propose a graph based approach for referring expression generation [16]. This approach allows to express relationships between objects (for example spatial relationships) in addition to each object’s attributes in a single model. We use a similar graph in our work but with uncertain attributes.

Our work is also an extension of previous work researching attribute detection and description generation. For example, Farhadi et al. [5] detect attributes of objects in scene, and use them as a description. The initial description includes all attributes and results in a lengthy description. With no task in mind, they are not able to measure the usefulness of the description. In our work, which is task specific, we are able to select attributes and anchors in a smart way, and show the utility of our descriptions.

Kumar *et al.* have performed in-depth research on nameable attributes for human faces. These attributes can be used for face verification and image retrieval [18], and similarity search [23]. Although we use the same attributes used in these works we utilize them in an entirely different way. Instead of using attributes as queries from a user to the al-

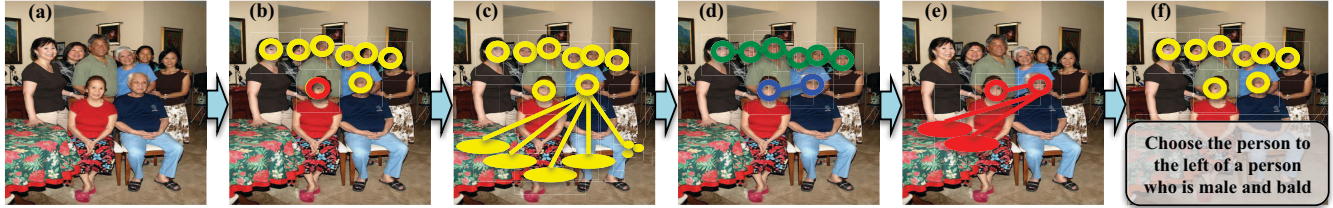


Figure 2. An overview of our algorithm. (a) Given an image of a group of people (b) detect all faces and select a random target. (c) For each face run a set of attribute classifiers. (d) Select neighbors by detecting rows of people. (e) Find a small set of attributes which refers to the target face with confidence c and use an anchor face if necessary (e) Construct a sentence and present to a guesser.

gorithm, or as features to calculate similarity, we use them as a method of communication from the computer to a user. This goal requires an efficient method for attribute selection which we propose here.

In recent years, some work has automatically composed descriptions of entire scenes. Although this is different from describing a specific object within a scene, there are similarities. For example, Berg et al. [1] predict what is important to mention in a description of an image by looking at the statistics of previous image and description pairs. They mention a few factors which can help predict if an item will be mentioned in a description such as size, object type and unusual object-scene pairs.

Both Farhadi et al. [6] and Ordonez et al. [20] find a description from a description database that best fits the image. Gupta et al. [13] use a similar approach, but instead break the descriptions into phrases to achieve more flexibility. Kulkarni et al. [17] use a CRF to infer the objects, attributes and spatial relationships which exist in a scene, and then compose all of them into a sentence. The main difference between this line of work and ours is the fact that our description is goal-oriented. That is, since these works produce a general description, they focus solely on the information within the scene. In contrast, we consider attribute scores for all objects, and the spatial relationship between them to describe the target object (person) in a way that discriminates him from others.

Finally, Sadovnik et al. [21] produces referring expressions for entire scenes. However, our method differs in major ways. First, [21] ranked various attributes, but they did not provide a way to calculate how many should be used. In our method, we calculate the necessary description length. Second, they do not rigorously deal with the uncertainty of the attribute detectors. They heuristically penalize for low confidence, but our formulation more naturally considers uncertainty. Finally, creating referring expressions for objects in a scene as opposed to entire scenes is more natural and has more practical applications (as described in Sec. 1).

In summary, our contribution is to produce a referring expression for a person in an image, and introducing anchor neighbors when the person is not sufficiently distinguished from others in the image. The rest of the paper is structured as follows. Sections 2 and 3 describe the Guesser Based

Model (GBM) algorithm from our previous work [22] (see a summary of the algorithm in Fig. 2). Section 4 discusses the results of our initial experiments and our extensions of using neighbors in the descriptions. Finally, Section 5 presents our new results and shows that using neighbors can significantly improve our descriptions.

2. Attributes and Neighbors

2.1. Attribute detection

Although the description algorithm we present is general, we choose to work with people attributes because of the large set of available attributes. Kumar et al. [18] define and provide 73 attribute classifiers via an online service. We retain 35 of the 73 attributes by removing attributes whose classification rate in [18] is less than 80%, and remove attributes which are judged to be subjective (such as attractive woman) or useless for our task (color photo). In the future other attributes (such as clothing, pose, or location in the image) can be easily incorporated into this framework.

Each classifier produces an SVM classification score for each attribute. Since our method requires knowledge about the attribute’s likelihood, we normalize these scores. We use the method described in [25] which fits an isotonic function to the validation data. We first collect a validation set for our 35 attributes, and fit the isotonic function using the method described in [2].

2.2. Neighbor Detection

A certain person might not have enough distinctive attributes to separate him from others in the group. Therefore, we wish to be able to refer to this person by referring to people around him. However, deciding who is standing next to whom is not a trivial task. We use the work of Gallagher et al. [8], to identify specific rows of people in a group photo.

We use this information to define faces who have a common edge in a row as neighbors. This gives us the “to the left of” and “to the right of” relationships. Since in [8] faces can be labeled as in the same row even though they are far apart, we add an additional constraint which normalizes the distance between every two faces in a row by the size of the face, and removes edges where the normalized size is

Variable Name	Variable Description
n	Number of people
$f \in \{1, 2, \dots, n\}$	Person to be described
\mathbf{A}	Set of binary attributes
$\mathbf{a}^* = [a_1^*, a_2^*, \dots, a_q^*]$ $a_k^* \in \mathbf{A}$	The attributes chosen by the algorithm for description
$\mathbf{v}^* = [v_1^*, v_2^*, \dots, v_q^*]$ $v_k^* \in \{0, 1\}$	Values chosen by the algorithm for the attributes in \mathbf{a}^*
$\mathbf{p}_k = [p_{k1}, p_{k2}, \dots, p_{kn}]$ $k = 1 \dots q$ $p_{ki} \in [0, 1]$	Probability of attribute k as calculated by classifier for each person
$\mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{kn}]$ $k = 1 \dots q$ $x_{ki} \in \{0, 1\}$	Values of attribute k of \mathbf{a}^* as seen by the guesser
$\tilde{f} \in \{1, 2, \dots, n\}$	Guesser's guess
$P_{\tilde{f}} = P(\tilde{f} = f \mathbf{a}^*, \mathbf{v}^*)$	The probability of the guesser guessing correctly
$t = \sum_{i=1}^n (x_{ki} == v_k^*)$	Number of faces with correct attribute value

Table 1. Variable definition

greater than some threshold t . This prevents distant people from being considered neighbors.

3. Guesser Based Model

As stated in Sec. 1 the goal of a referring expression generator is to find a short description that refers to a single object in the scene. In order to decide when anchor faces are needed we must develop a method to calculate a probability of a user guessing correctly. Calculating this probability relies on a guesser model which is provided in Sec. 3.1.

We then describe how to calculate the probability that the guesser will, in fact, guess the target face given any description within the space of our attributes by considering the uncertainty of the attribute classifiers. First, we explain this calculation when the description has a single attribute (Sec. 3.2). Then, we explain the extension to the case when the description contains multiple attributes (Sec. 3.3). In both cases, we show that this calculation is polynomial in both the number of faces in the image, and the number of attributes in the description.

Finally, we describe the algorithm for producing attribute descriptions that meet our goals: having as few attributes as possible, while selecting enough so that that probability of a guesser selecting the the target person will be higher than some threshold. We also describe a method for selecting an anchor face to use if we cannot reach the threshold by just using the target's attributes.(3.4).

3.1. Guesser's Model

We first define a model that the guesser follows to guess the identity of the target person, given an attribute description. All variables are defined in Table 1. Given that he has received a set of attribute-value pairs (\mathbf{a}^* , \mathbf{v}^*), he guesses the target face \tilde{f} according to the following rules:

- If only one person matches all attribute-value pairs guess that person.

Classifier's Probabilities						
Smiling	0.8	0.4	0.2			
x_k	Face 1	Face 2	Face 3	Prob. of happening	Prob. of guessing correct	Prob. of happening and of guessing correct
[1,1,1]	😊	😊	😊	$0.8*0.4*0.2$	0.333	0.021
[1,0,0]	😊	😞	😞	$0.8*0.6*0.8$	1	0.384
[0,0,0]	😞	😞	😞	$0.2*0.6*0.8$	0.333	0.032
⋮						⋮
Probability of guessing correct: $0.021 + 0.384 + 0.032 + \dots + 0 = 0.613$						

Figure 3. An illustration calculating the probability of guessing correctly using one attribute (“The person is smiling”) for an image with three people. The true identity of the target person (marked with a red rectangle) is known to the algorithm as well as the attribute confidence for each face. Each face is actually smiling or not (the true state is unknown to the algorithm), represented with the blind over each mouth. To find the probability of the guesser’s success, each of the eight possible configurations of smiling faces is considered.

- If more than one person matches all attribute-value pairs guess randomly among them.
- If no person matches any attribute-value pairs guess randomly among all people.
- If no person matches all attribute-value pairs, choose randomly among the people who have the most matches.

Given this model, the describer’s goal is to maximize $P_{\tilde{f}} = P(\tilde{f} = f | \mathbf{a}^*, \mathbf{v}^*)$, the probability that the guesser correctly identifies the target, given the description. Following Grice’s Maxim of Quantity we also wish to create a short description. Therefore, we choose to explore descriptions that minimize the number of attributes $|\mathbf{a}^*|$ such that $P_{\tilde{f}} > c$, where c is some confidence level.

In order to illustrate how $P_{\tilde{f}}$ is calculated we first present the single attribute case, and then extend to multiple attributes.

3.2. Single Attribute

We now formalize our algorithm. Here, for simplicity of notation, the description is comprised of positive attributes (e.g., “the smiling face”), but we also consider negative attributes (e.g., “the face that is not smiling”) by taking the compliment of the attribute probability scores for each face. The probability of each possible \mathbf{x}_k occurring is:

$$P(\mathbf{x}_k) = \prod_{i=1}^n (x_{ki}p_{ki} + (1 - x_{ki})(1 - p_{ki})) \quad (1)$$

For each \mathbf{x}_k and attribute-value pair (a_k^* , v_k^*) we compute the probability of the guesser guessing correctly using the guesser model:

$$P(\tilde{f} = f | \mathbf{x}_k, a_k^*, v_k^*) = \begin{cases} \frac{1}{n} & \text{if } t = 0 \\ 0 & \text{if } x_{kf} = 0 \text{ \& } t > 0 \\ \frac{1}{t} & \text{otherwise} \end{cases} \quad (2)$$

Therefore, we calculate the total probability of a correct guess given a single attribute by summing over all (2^n) configurations of the attribute over the faces in the image as (see example in Fig 3):

$$P_{\tilde{f}} = \sum_{\mathbf{x}_k} P(\tilde{f} = f | \mathbf{x}_k, a_k^*, v_k^*) P(\mathbf{x}_k) \quad (3)$$

In Eq. 3, we sum over all possible \mathbf{x}_k which is exponential in the number of faces n and computationally expensive. Since the images in our dataset contain many faces, it is intractable. However, we notice that $P_{\tilde{f}}$ depends only on the number of faces t that satisfy the attribute, given that the target face does. We can rewrite Eq. 3 as:

$$P_{\tilde{f}} = \frac{1}{n} P(t = 0) + 0 + \sum_{\mathbf{x}_k | x_{kf} = 1} \frac{1}{t} P(\mathbf{x}_k) \quad (4)$$

Where each of the three terms in the sum refer to the three terms in Eq. 2 respectively. We notice that t is actually a Poisson-Binomial random variable whose PMF (probability mass function) can be computed in time polynomial with the number of faces. A Poisson-Binomial distribution is the distribution of the sum of independent Bernoulli trials where the parameter p can vary for each trial (as opposed to the Binomial distribution). We can calculate the PMF efficiently by convolving the Bernoulli PMF's [7]. In our case, the parameters of the random variable are p_k . We can therefore rewrite Eq. 4 as:

$$P_{\tilde{f}} = \frac{1}{n} P(t = 0) + 0 + p_{kf} \sum_{t=1}^n \frac{1}{t} P(t | x_{kf} = 1) \quad (5)$$

Since inside the summation we only care about cases in which $x_{kf} = 1$ we set the Poisson-Binomial parameter for face f to 1 and then compute the PMF of t . Eq. 5 provides a way to calculate the value of Eq. 3 exactly while avoiding the summation over all possible \mathbf{x}_k . We can now compute $P_{\tilde{f}}$, the probability that the guesser will succeed, in time polynomial with the number of faces.

Using Eq. 5 we can find, from a pool of available attributes, the single best attribute to describe the target face (the a_k^*, v_k^* that maximizes $P_{\tilde{f}}$). Extending this strategy to multi-attribute descriptions is not trivial. One greedy algorithm for producing a multi-attribute description is to order all available attributes by $P_{\tilde{f}}$, and choose the top m . However, this could yield redundant attributes. For example, imagine a group photo with two people who both have glasses and are senior, one of whom is our target. The attribute-value pairs *has glasses* and *is senior* may be the top two with the greatest $P_{\tilde{f}}$. However, mentioning both

attributes is useless, because they do not contain new information. What is actually needed is a method of evaluating the guesser success rate with a multi-attribute description.

3.3. Multiple Attributes

We introduce a new random variable y_i , the number of attributes of face i which match the description $(\mathbf{a}^*, \mathbf{v}^*)$.

$$y_i = \sum_{j=1}^q x_{ji} == v_j^* \quad (6)$$

y_i is also a Poisson-Binomial random variable whose parameters are $p_{ji} \mid j = \{1, 2 \dots q\}$. We expand the definition of t from our single attribute example. Whereas previously it signified the number of faces with the correct value for a single attribute, t_j now signifies the number of faces with exactly j matching attributes.

$$t_j = \sum_{i=1}^n y_i == j \quad (7)$$

Using these random variables we efficiently calculate the guesser's success given multiple attributes. The basic idea is to look at the case when the target face has j correct attributes and no other face has more than j attributes correct (if any other face does the probability of guessing correctly is zero), and then perform Eq. 5 using t_j where our new p values are the probabilities of having j attributes normalized by the probability of having j or less attributes. Summing over all values of j gives us the following equation:

$$P_{\tilde{f}} = \sum_{j=1}^q \sum_{t_j=1}^n \left(\frac{1}{t_j} p(t_j | y_f = j, y_i \leq j \forall i) \right. \\ \left. \times p(y_f = j | y_i \leq j \forall i) p(y_i \leq j \forall i) \right) \quad (8)$$

3.4. Guesser-Based Attribute Selection

We perform attribute selection in a similar fashion to the Greedy Heuristic Method. The algorithm's pseudo code is shown in Algorithm 1. In this greedy method, at each step we select the best attribute-value pair to add to our current solution that gives the highest combined probability of guessing correctly given our selection from the previous step (evaluated with Eq. 8).

As mentioned in Sec. 2.2, we can use neighboring people to anchor the description when needed. If we cannot create a target's description with a confidence level above a certain threshold, we look at each of the target's neighbors. For each neighbor, we rerun the algorithm using both the target's and the neighbor's attributes, doubling the number of attributes we can choose from. That is, in this case we look for a set of attributes that would differentiate this pair of people from all other pairs. This allows us to create referring statements such as "The person with the glasses to left

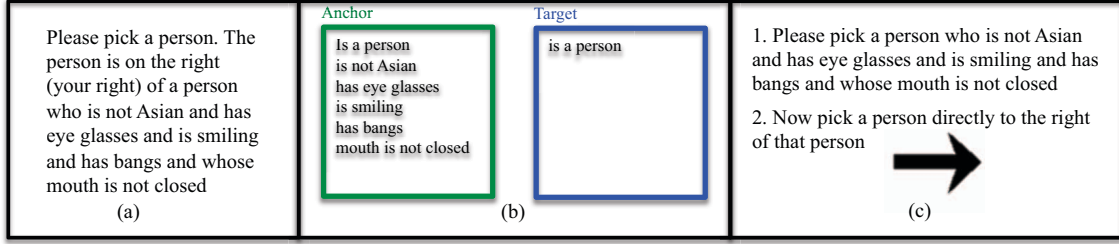


Figure 4. Examples of 3 different ways we presented our descriptions. (a) Text: an exclusively textual description as in [22]. (b) Graphical: Our graphical representation (c) Two-Step: Our two step presentation. The second part is only shown after the first part was completed.

of the person with the beard” that are effective even when there are other distractors with glasses and beards, so long as they are not standing in that specific layout.

Algorithm 1: Attribute selection algorithm

```

Data:  $c, A, f$ 
Result:  $a^*, v^*$ 
1  $a^* \leftarrow \emptyset;$ 
2  $curr\_conf \leftarrow 0;$ 
3 while ( $curr\_conf < c$ ) do
4   for each  $A_i \notin a^*$  do
5      $tmp\_A \leftarrow a^* \cup A_i;$ 
6     for each  $tmp\_v$  do
7       calculate  $p = P(\tilde{f} = f | tmp\_A, tmp\_v);$ 
8       if  $p > curr\_conf$  then
9          $curr\_conf \leftarrow p;$ 
10         $curr\_best \leftarrow (tmp\_A, tmp\_v)$ 
11      end
12    end
13  end
14   $(a^*, v^*) \leftarrow curr\_best$ 
15 end

```

Once we have a set of attributes we construct a sentence. Since the main focus of this paper is on the selection method we use a simple template model to build the sentences.

4. Preliminary Results

To examine the effectiveness of our algorithm’s descriptions, we run a set of Amazon Mechanical Turk (AMT) experiments in which a user must choose a face according to a given description (See Sec. 5 for more details). The results we achieved in [22] show that without using neighbors the GBM method performs better than all baselines. However, although GBM_neighbors, which allows using neighbors as described in Sec. 3.4, had a higher average predicted confidence level (0.82) than GBM (0.65), it produced lower guessing results (52% vs. 59% respectively).

We hypothesized two main reasons why the results achieved by GBM_neighbors were worse. Our first hypothesis was that although the information in the description should yield higher guessing results, the sentence itself was unclear, and was presented in a way that confused the guesser. For example, the user might have been confused

about the direction (right vs. left), or confused about who to select (anchor face vs. target face).

In order to test our first hypothesis, we created a new presentation using a graphical diagram instead of the textual description. An image with two squares was presented to the user (Fig. 4(b)), one labeled target and the other labeled anchor, and within each square the relevant attributes were listed. We then required the user to select both the anchor and the target face. We believed that this graphical representation would solve the confusion of left and right, and in addition, by forcing the user to select the anchor, we could better analyze the error types.

Our second hypothesis was that people were having a hard time finding the correct person since both the target and the anchor face were described as a unique pair. That is, when choosing the attributes to include in the description, we allow the algorithm to try ones from both the target and anchor face. Therefore, although the description refers to this pair with high confidence, it requires a comparison to all other pairs which might prove too difficult for the average Mechanical Turk user.

In order to test our second hypothesis, we created a new type of description: GBM_neighbors*. In this model, if we cannot create a description with a confidence above the threshold for just the target we look at the target’s neighbors individually, and choose the description with the highest confidence. That is, these descriptions will only include attributes from one anchor person as opposed to GBM_neighbors which allowed selecting attributes from both. If a neighbor’s description has a higher confidence, we simply request the user to select the person to the left/right of the described anchor person. Although this model produces lower confidences than GBM_neighbors (0.77 vs. 0.82), it creates a description of a single person which, according to our second hypothesis, is clearer.

Using GBM_neighbors* allowed us to try a different presentation. Since the anchor face is the only one described with attributes, the user could guess iteratively. First the user is asked to select the anchor face only. This task is the same as the testing performed on our regular GBM model. Once a face is selected the user is prompted to select an additional face to the left/right of the first selected face. In order to clarify the direction we present an arrow (Fig. 4(c)).

	Graphical GBM_neighbors		Graphical GBM_neighbors*		Two-step GBM_neighbors*	
	True Anchor	True Target	True Anchor	True Target	True Anchor	True Target
Guessed Anchor	42.0%	9.5%	46.8%	7.6%	64.3%	3.1%
Guessed Target	10.8%	30.6%	7.8%	31.1%	3.3%	55.7%
Sum	52.8%	40.1%	54.6%	38.7%	67.6%	58.8%

(a) (b) (c)

Table 2. Results of our three different experiments as described in Sec. 5. (a) and (b) use the presentation method as shown in Fig. 4(b), while (c) uses the presentation method as shown in Fig. 4(c). The last row is the sum of the first two, and signifies the total percentage of people who chose the true target/anchor as one of their choices.




			
GBM	Pick a person who has bangs and whose forehead is not fully visible and whose teeth are visible and is wearing lipstick and is not black 1/3	Pick a person who does not have black hair and does not have eye glasses and is chubby and is smiling and whose teeth are visible 1/3	Please pick a person who is a male and is in their youth and has black hair and does not have eye glasses and does not have a mustache 1/3
GBM_Neighbors*	Pick a person. The person is on the left (your left) of a person who has a mustache and has a beard and whose teeth are not visible and is not black 4/4	Pick a person. The person is on the right (your right) of a person who is a male and has black hair and whose forehead is not fully visible and does not have a mustache and whose teeth are visible 4/4	Pick a person. The person is on the left (your left) of a person who is a child and is not middle aged and has black hair and whose mouth is closed and whose teeth are not visible 4/4

Figure 5. Examples of the different descriptions created using GBM vs GBM_neighbors*, and the accuracy achieved in our collected results. In these examples it is clear to see that since it is hard to differentiate the target person from the distractors, using a neighbor anchor face clearly simplifies the task.

5. Experiments and Results

To examine our different descriptions and presentations we follow the same experimental method used in [22]. We use images from the Images Of Groups Dataset [9] that contain at least 8 people. We create 1200 descriptions for 400 faces (GBM, GBM_neighbors, GBM_neighbors*). Since our focus is on the differences between these algorithms, we only run our experiments on the 165 faces for which neighbors were used. On the additional 235 faces, all three algorithms produced the same description, and therefore no difference would be observed.

We evaluate our algorithm with experiments on AMT. We present a worker with an image with all detected faces marked with a square and a description (either textual or graphical as described in Sec. 4), and ask them to select who is being referred to. The selection is done by clicking on a face. Each worker performs a random set of ten image-description pairs with one guess each. We encourage the workers to guess correctly by offering a monetary bonus to the top guessers. On average, three separate AMT workers guess each image. We set our confidence level c to 0.9 and the maximum number of attributes to 5. For faces which do not reach confidence level c , we use the description with the highest score with at most 5 attributes.

We use the original GBM and GBM_neighbors descriptions from [22] on our new dataset which achieved

41.47% and 36.6% accuracy respectively. These results are inline with previous results which show that using GBM_neighbors decreases the guessing accuracy. The lower overall performance is expected since we are only looking at the 165 faces for which the confidence score was below the threshold for GBM.

We next tried our graphical representation as shown in Fig. 4(b). In these experiments we asked the users to select the anchor face as well, and so had greater insight into errors. Table 2(a) presents a confusion matrix of guessed/true anchor/target faces. The columns do not add up to 1 since many faces selected were neither the target nor the anchor.

When looking at the target guessing accuracy (30.6%), we observed an actual decrease from the textual presentation of the GBM_neighbors description. However, when adding up the number of true targets guessed as anchors, we observe an accuracy increase (40.1%), indicating confusion about whether the worker should select the anchor face, or the target face.

Our next experiment presented descriptions created by GBM_neighbors* in the same graphical format 4(b). Since the description given to the anchor face selected by the algorithm will definitely have a higher confidence, we predicted that at least the guessing rate for the true anchor will be higher than that for the target of the GBM algorithm. Results are presented in Table 2(b). Although the guessing rate for the true anchor had improved as expected

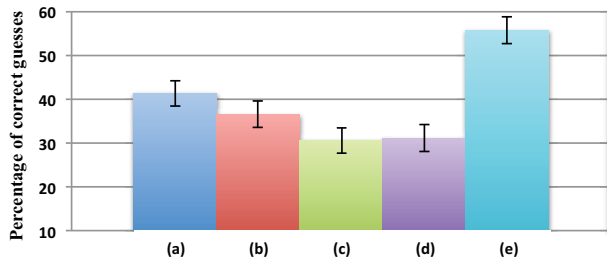


Figure 6. Our final results. (a) Text GBM (b) Text GBM_neighbors(c) Graphical GBM_neighbors (d) Graphical GBM_neighbors* (e) Two-step GBM_neighbors*

(46.8%), the target guessing accuracy remained comparable to GBM_neighbors.

This motivated our two-step presentation method (Fig. 4(c)). We reasoned that if people are able to guess the anchor face with higher accuracy, then the main problem was still with understanding where the target face is in relation to it. This new presentation method breaks the task into two steps and clarifies the exact direction in which the additional face needs to be chosen. Table 2(c) presents the results of this experiment. It is important to note that this type of iterative description would not work for GBM_neighbors, since that method describes the pair jointly and cannot be reduced to two independent selection tasks.

As predicted, this final combination of GBM_neighbors in addition to our two step presentation method performs the best on both target and anchor faces. The higher accuracy for the anchor face vs. the target face is to be expected since getting the anchor correct does not guarantee guessing the target even if the direction is clear (it can be ambiguous who is exactly to the right of a person).

Fig. 5 shows examples in which using neighbors clearly helps, while Fig. 6 shows the target guessing accuracy for all 5 experiments conducted. From our final results it is clear to see that while we can clearly improve the results using neighbors ((e) vs. (a)), it depends on how we use the neighbors and the presentation method.

6. Conclusion

We have shown that using neighbors as anchors in referring expressions can significantly improve their accuracy. Although from our initial experiments we observed that using neighbors can hurt guesser’s accuracy since it makes the expression more complicated, we have shown that if presented correctly it can yield positive results.

An interesting direction for future research could be further investigating different types of presentations. Although we present evidence that an effective presentation of referring expressions with neighbors is necessary to achieve higher results, the most efficient way to present this type of description remains an open question.

References

- [1] A. Berg, T. Berg, H. Daume, J. Dodge, A. Goyal, X. Han, A. Mensch, M. Mitchell, A. Sood, K. Stratos, et al. Understanding and predicting importance in images. In *CVPR*, 2012. 3
- [2] O. Burdakov, O. Sysoev, A. Grimvall, and M. Hussian. An $o(n^2)$ algorithm for isotonic regression. *Large-Scale Nonlinear Optimization*, pages 25–33, 2006. 3
- [3] R. Dale. Cooking up referring expressions. In *ACL*. Association for Computational Linguistics, 1989. 2
- [4] R. Dale and E. Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995. 2
- [5] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2
- [6] A. Farhadi, S. M. M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010. 3
- [7] M. Fernandez and S. Williams. Closed-form expression for the poisson-binomial probability density function. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(2):803–817, 2010. 5
- [8] A. Gallagher and T. Chen. Finding rows of people in group images. In *ICME*, 2009. 3
- [9] A. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009. 7
- [10] A. Gatt, I. Van Der Sluis, and K. Van Deemter. Corpus-based evaluation of referring expression generation. *Position Papers*, 2007. 2
- [11] D. Golland, P. Liang, and D. Klein. A game-theoretic approach to generating spatial descriptions. In *Conference on empirical methods in natural language processing*, 2010. 2
- [12] P. Grice. Logic and conversation. *Syntax and Semantics*, 3:43–58, 1975. 1
- [13] A. Gupta, Y. Verma, and C. Jawahar. Choosing linguistics over vision to describe images. In *AAAI*, 2012. 3
- [14] H. Horacek. Generating referential descriptions under conditions of uncertainty. In *ENLG*, 2005. 2
- [15] J. D. Kelleher and G.-J. M. Kruijff. Incremental generation of spatial referring expressions in situated dialog. In *ACL*, 2006. 2
- [16] E. Kraemer, S. Erk, and A. Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 2003. 2
- [17] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011. 3
- [18] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. In *PAMI*, Oct 2011. 2, 3
- [19] C. Mellish, D. Scott, L. Cahill, D. Paiva, R. Evans, and M. Reape. A reference architecture for natural language generation systems. *Natural Language Engineering*, 12(01):1–34, 2006. 1
- [20] V. Ordonez, G. Kulkarni, and T. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 3
- [21] A. Sadovnik, Y. Chiu, N. Snaveley, S. Edelman, and T. Chen. Image description with a goal: Building efficient discriminating expressions for images. In *CVPR*, 2012. 3
- [22] A. Sadovnik, A. Gallagher, and T. Chen. It’s not polite to point: Describing people with uncertain attributes. In *CVPR*, 2013. 1, 2, 3, 6, 7
- [23] W. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012. 2
- [24] J. Viethen and R. Dale. The use of spatial relations in referring expression generation. In *International Natural Language Generation Conference*, 2008. 2
- [25] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD*, 2002. 3