2013 IEEE Conference on Computer Vision and Pattern Recognition

It's Not Polite To Point: Describing People 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

Visual attributes are powerful features for many different applications in computer vision such as object detection and scene recognition. Visual attributes present another application that has not been examined as rigorously: verbal communication from a computer to a human. Since many attributes are nameable, the computer is able to communicate these concepts through language. However, this is not a trivial task. Given a set of attributes, selecting a subset to be communicated is task dependent. Moreover, because attribute classifiers are noisy, it is important to find ways to deal with this uncertainty. We address the issue of communication by examining the task of composing an automatic description of a person in a group photo that distinguishes him from the others. We introduce an efficient, principled method for choosing which attributes are included in a short description to maximize the likelihood that a third party will correctly guess to which person the description refers. We compare our algorithm to computer baselines and human describers, and show the strength of our method in creating effective descriptions.

1. Introduction

Imagine you are at a party with many people, and need to point out one of them to a friend. Because it is impolite to point (and it is difficult to follow the exact pointing direction in a large group), you describe the target person to your friend in words. Most people can naturally decide what information to include in what is known in the Natural Language Processing field as a *referring expression*. For example, in Figure 1, we might say: (a) "The man who is not wearing eyeglasses" (b) "The man who is wearing eyeglasses" or (c) "The woman".

The task of generating these expressions requires a balance between the two properties of Grice's Maxim of Quantity [10]. The maxim states:

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

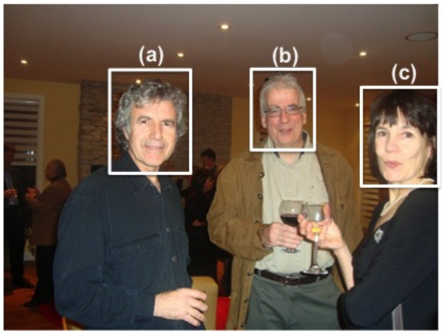


Figure 1. In this paper we introduce an efficient method for choosing a small set of noisy attributes needed to create a description which will refer to only one person in the image. For example, when the target person is person *(b)*, our algorithm produces the description: "Please pick a person whose forehead is fully visible and has eyeglasses"

In our context, in which the computer attempts to refer to a single person, we interpret these as follows. First, the description ideally refers to only a 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.

Although people find this describing task to be easy, it is not trivial for a computer. First, computers must deal with uncertainty. That is, the attribute classifiers the computer uses are known to be noisy and this uncertainty must be considered in an effective model. In addition, given that each person in our image might have many attributes describing him, selecting the smallest set of attributes with which to describe him uniquely is an *NP-hard* problem[4]. For example, a brute-force method is to first try all descriptions with one attribute, then try all descriptions with two attributes and so on. Although this will find the shortest description, the computational complexity is exponential in the number of available attributes.

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 [18].

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.397

When giving a general description one might be required to refer to specific objects within the scene. For example in Figure 1, we might say "The person wearing eyeglasses is the company's president," instead of simply "The person is the company's president." This type of referral is crucial in generating informative image captions. Our algorithm provides a method for selecting which attributes should be mentioned in such a case.

This research has practical applications. In security, surveillance cameras and action recognition algorithms can identify suspicious people. A security guard could receive concise verbal descriptions of the suspect to investigate. Both properties of the description are extremely crucial. First, the description needs to refer only to the suspect to prevent investigating the wrong person. Second, it must not be too long as to confuse the guard or waste his time.

Another application involves navigation systems. Using a front-facing camera on a car and a GPS system, we can develop a system which can provide more intuitive driving directions. For example, instead of saying: "Turn right in 200 feet," it might be more useful to say: "Turn right at the yellow building with the red awning," or even "Follow the green car that just turned right." Although we use our algorithm for describing people, it is is not confined to this specific domain. By employing object detection algorithms, in addition to other attribute classifiers, a general system can be realized.

Our main contributions are: We present the first attempt at generating referring expressions for objects in images. This task has been researched in the NLG community, but had yet to use visual data with actual uncertainties. In addition, we present a novel and computationally efficient method for evaluating the probability that a given description will result in a correct guess from the listener. Finally, we develop a new algorithm for attribute selection which takes into consideration the uncertainty of the classifiers. That is, although we cannot guarantee that the description we compose will describe only the target person, we are able to select attribute combinations for a high probability of this occurring.

1.1. Previous Work

There has been active computational research on referring expression generation in the NLG community for 20 years. Most consider 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. We build on this work from a computer vision point-of-view, using actual attribute predictions made from analyzing real images of people.

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 additional 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 real attribute classifiers, and introduce an efficient algorithm for dealing with this uncertainty.

Other extensions to these three main algorithms have been proposed. For example, Krahmer *et al.* propose a graph base approach for referring expression generation [14]. The reason for using this approach is that it allows for relationships between objects to be expressed (for example spatial relationships) in addition to the individual attributes of each object. We use a similar graph in our work.

Horacek proposes an algorithm which deals with conditions of uncertainty [12]. This method is similar to the one we are proposing 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 [12], we provide experimental data to show our algorithm's strength.

Although this is the first attempt at generating referring expressions for objects in images, our work is 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 in a smart way, and show the utility of our descriptions.

Attributes improve object classification [17, 20] and search results [13]. For example, Kumar et al. describe in-depth research on nameable attributes for human faces. These attributes can be used for face verification and image retrieval [16], and similarity search [22]. These works all use human-generated attribute feedback to help a computer at its task. In contrast, in our case the computer (not a human) is the one generating descriptive attribute statements, so the emphasis is on selecting attributes, even when the classifier scores are uncertain.

In recent years, attributes have been used to automatically compose 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

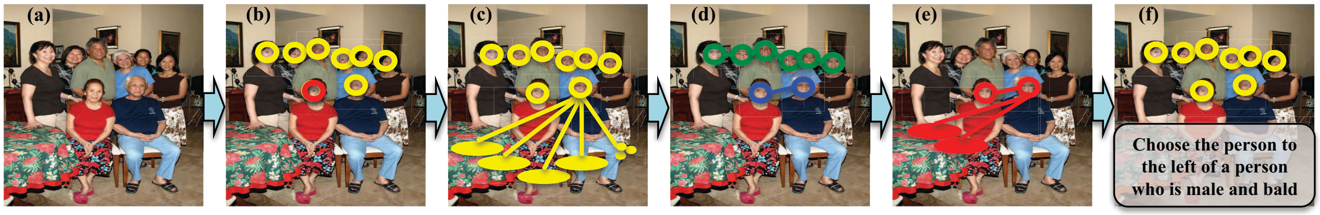

(a)
(b)
(c)
(d)
(e)
(f)
Choose the person to the left of a person who is male and bald

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. (d) Find a small set of attributes which refers to the target face with confidence c (e) Construct a sentence and present to a guesser.

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 (e.g., size, object type and unusual object-scene pairs) to help predict whether an item will be mentioned in a description.

Both Farhadi et al. [6] and Ordonez et al. [19] find a description from a description database that best fits the image. Gupta et al. [11] use a similar approach, but break descriptions into phrases to realize more flexible results. Kulkarni et al. [15] use a CRF infer objects, attributes and spatial relationships that exist in a scene, and 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, prior works focus solely on the information and scores within the scene. In contrast, we consider attribute scores for all objects 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 improves on [21] in major ways. First,[21] ranked various attributes, but did not provide a calculation of how many attributes should be used. In our method, we calculate the necessary description length. Second, we rigorously deal with the uncertainty of the attribute detectors, instead of using a heuristic penalty for low confidence as in [21]. 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).

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. [16] define and provide 73 attribute classifiers via an online service. We retain 35 of the 73 attributes by removing attributes whose classification rate in [16] is less than 80%, and removing attributes which are judged to be subjective (such as attractive woman) or useless for our task (color photo). In the future other attributes can be easily incorporated into this framework such as clothing or location in the image.

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

3. Algorithm

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 our scenario of uncertain classifiers, our goal is to produce a description that will allow a guesser a high probability of successfully guessing the identity of the target face. Calculating this probability relies on a guesser model which we provide in Sec. 3.1. The guesser model defines the strategy used by the listener to guess which face in the image is the one being described.

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

Variable Name	Variable Description
n	Number of people
$f \in \{1, 2, \ldots, n\}$	Person to be described
$\mathbf{A}$	Set of binary attributes
$\mathbf{a}^* = [a_1^*, a_2^*, \ldots a_q^*]$ $a_k^* \in \mathbf{A}$	The attributes chosen by the algorithm for description
$\mathbf{v}^* = [v_1^*, v_2^*, \ldots, v_q^*]$ $v_k^* \in \{0, 1\}$	Values chosen by the algorithm for the attributes in a^*
$\mathbf{p_k} = [p_{k1}, p_{k2}, \ldots, p_{kn}]$ $k = 1 \ldots q$ $p_{ki} \in [0, 1]$	Probability of attribute k as calculated by classifier for each person
$\mathbf{x_k} = [x_{k1}, x_{k2}, \ldots, x_{kn}]$ $k = 1 \ldots q$ $x_{ki} \in \{0, 1\}$	Values of attribute k of a^* as seen by the guesser
$\tilde{f} \in \{1, 2, \ldots, n\}$	Guesser's guess
$P_{\tilde{f}} = P(\tilde{f} = f \vert \mathbf{a}^*, \mathbf{v}^*)$	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 definitions

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 introduce an 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 (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:

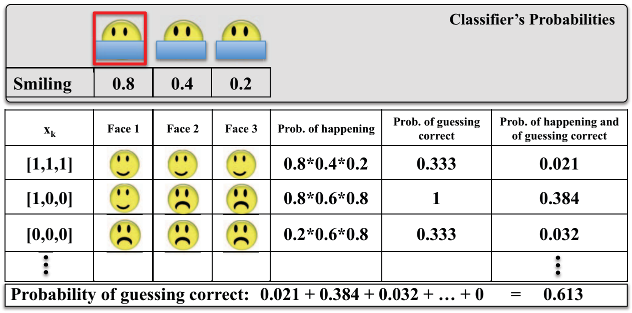
- If only one person matches all attribute-value pairs guess that person.
- If more than one person matches all attribute-value pairs guess randomly among them.
- If no person matches any attribute-value pairs guess randomly among all people.
- If no person matches all attribute-value pairs, choose randomly 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 $|a^*|$ such that $P_{\tilde{f}} > c$, where c is some confidence level.

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

3.2. Single Attribute

Consider the case where a "smile detector" is applied to an image containing three faces, and we refer to face 1 as "the smiling face" (Figure 3). What is the probability that a guesser will be correct? To compute this, we must consider the fact that our smile detector is never certain, but instead, reports confidences of observing a smile on each face. The confidence associated with each score represents the probability that each face actually has a smile or not. The actual joint distribution of smiling faces in the image has eight possibilities over the three faces (2^3). For each of these eight possible arrangements, the probability that the guessing strategy leads to a correct guess can be computed. Naïvely, by applying total probability, the overall probability of guesser success is the sum of the probability that each of these eight smile cases occur, times the probability of guesser success in each case.



Classifier's Probabilities

Smiling	0.8	0.4	0.2

$\mathbf{x_k}$	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 + … + 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. We introduce a polynomial-time algorithm for computing this probability.

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:

$$P(\mathbf{x_k}) = \prod_{i=1}^{n} (x_{ki}p_{ki} + (1 - x_{ki})(1 - p_{ki})) \qquad (1)$$

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:

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

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:

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

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:

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

Where each of the three terms in the sum refer to the three terms in Eq. 2 respectively. Finally, 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 Bernouli PMF's [7]. In our case, the parameters of the random variable are p_k . We can therefore rewrite Eq. 4 as:

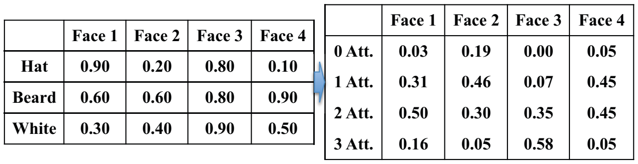
$$P_{\tilde{f}} = \frac{1}{n}P(t=0) + 0 + p_{kf}\sum_{t=1}^{n} \frac{1}{t}P(t|x_{kf}=1) \tag{5}$$

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 ploynomial 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 correctly match the description $(\mathbf{a}^*, \mathbf{v}^*)$.



	Face 1	Face 2	Face 3	Face 4
Hat	0.90	0.20	0.80	0.10
Beard	0.60	0.60	0.80	0.90
White	0.30	0.40	0.90	0.50

	Face 1	Face 2	Face 3	Face 4
0 Att.	0.03	0.19	0.00	0.05
1 Att.	0.31	0.46	0.07	0.45
2 Att.	0.50	0.30	0.35	0.45
3 Att.	0.16	0.05	0.58	0.05

Figure 4. An example of transforming the table of p_{ki} into the 4 PMF's of y_i (one per column). In Eq. 8, j iterates through the different rows and normalizes accordingly.

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

In this work we consider all attributes to be independent. Therefore, y_i is also a Poisson-Binomial random variable whose parameters are $p_{ji} \bigm| j = \{1, 2 \dots q\}$ (as shown in Figure 4). 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.

$$t_j = \sum_{i=1}^{n} y_i == j \tag{7}$$

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 jth row of Figure 4 normalized by the sum of rows $0 - j$. Summing over all values of j gives us the following equation:

$$\begin{aligned} P_{\tilde{f}} = & \sum_{j=1}^{q}\sum_{t_j=1}^{n} \Bigg(\frac{1}{t^i} p(t_j|y_f = j, y_i \le j \ \forall i) \\ & \times p(y_f = j|y_i \le j \ \forall i)p(y_i \le j \ \forall i) \Bigg) \end{aligned} \tag{8}$$

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. This is a greedy method in which in each step we select the best attribute-value pair to add to our current solution, which gives us 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 when the target person does not have enough distinguishing attributes. We do this by setting an upper limit on the number of attributes used. If the algorithm fails to reach desired confidence , we re-run the algorithm using the neighbor's attributes as well. It should be emphasized that when using a neighbor we examine both sets of attributes jointly (that is, our attribute set is doubled). This allows us to create descriptions such as "The person with the glasses to left of the person with the beard".

Algorithm 1: Attribute selection algorithm

```
Data: $c, A, f$
Result: $a^*, v^*$
$a^* \leftarrow \emptyset$;
$curr\_conf \leftarrow 0$;
while ($curr\_conf < c$) do
    for each $A_i \notin a^*$ do
        $tmp\_A \leftarrow a^* \cup A_i$;
        for each $tmp\_v$ do
            calculate $p = P(\tilde{f} = f | tmp\_A, tmp\_v)$;
            if $p > curr\_conf$ then
                $curr\_conf \leftarrow p$;
                $curr\_best \leftarrow (tmp\_A, tmp\_v)$
            end
        end
    end
    $(a^*, v^*) \leftarrow curr\_best$
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 create a simple template model to build the sentences.

4. Experiments and Results

We perform two main experiments using Amazon Mechanical Turk (AMT). First we perform an evaluation of our algorithm by comparing it to a few baselines (Sec. 4.1). We also compare our algorithm's descriptions to ones we collect on AMT from human describers (Sec. 4.2).

4.1. Computer Baselines

To evaluate our algorithm we run experiments on AMT. Workers view an image with all detected faces marked with a square and a textual description, 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.

We compare the guessing accuracy for descriptions created using the following methods:

1. **Confident:** Compose the description from the n most confident attributes. This baseline completely ignores other faces in the image.

2. **Top_used:** After running the algorithm on the dataset, we select the n top used attributes throughout the whole set. The top 5 attributes are: gender, teeth visible, eyeglasses, fully visible forehead and black hair.

3. **Full_greedy:** We rank the attributes using the value of Eq. 5, skipping the method introduced in Sec. 3.3, and use the top n to compose the description.

4. **GBM:** Guesser Based Model. Our full algorithm without neighbors.

5. **GBM_neighbors:** Our algorithm with neighbors.

We create 2000 descriptions for 400 faces (1 for each method). These faces were randomly selected from all detections, and manually verified to be true detections. We have 3 separate AMT workers guess each, for a total of 6000 guesses. 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. For the rest of the algorithms, n is the number of attributes selected by GBM.

We use images from the Images Of Groups Dataset [9] that contain at least 8 people. The face detector detects 87% of the correct faces with 89% accuracy for an average of 11.4 faces per image (random guessing would achieve an average of 0.099). Results are presented in Figure 5. We also show description examples in Figure 6.

Examining the results, it is interesting that using the most confident attributes actually performs the worst, even worse than simply describing a constant set of attributes as in Top_used (P=0.0022). This shows that an attribute classifier score, by itself, is not enough information to construct an effective description for our task. Figure 5c hints at the reason for this. The attributes the classifier tends to be certain about are ones which are not useful for our task since they tend to be true for many people. For example, the *eyes open* attribute (8 in Figure 5c) is used in around 80% of the confident descriptions. However, this is usually not useful since most people have their eyes open. This fact is strengthened by the low usage of this attribute by the other methods.

The need to select attributes in a manner that takes into account the other faces in the image is clear from the improved performance when using our selection algorithms. Our Full_greedy approach reaches an accuracy of 56%. The additional 4% achieved when using GBM (P=0.0131) shows the improvement gained using the methods described in Sec. 3.3, which prevent mentioning redundant attributes (See Figure 7a for an example).

The fact that using neighbors lowered the accuracy surprised us since we were expecting an increase in accuracy. However, when examining the results carefully we observed some common errors which we believe led to this. First, since we only verified that the target face is not a false positive, there are no guarantees for the neighbors. Therefore, when a person is next to a false detection he may be referred to as standing *to the left of* that person which will obviously confuse the guesser. In addition, some people were confused by the reference and ended up choosing the neighbor

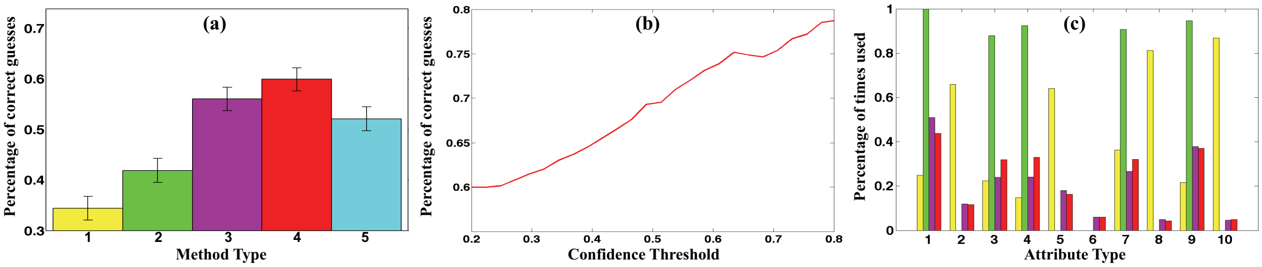


Figure 5. Our results from the computer baseline experiment (Sec. 4.1). (a) Guessing accuracies for the five methods introduced in Sec. 4.1. 1. confident 2. top_used 3. Full_greedy 4. GBM 5. GBM_neighbors. (b) Accuracy results of GBM as we increase the minimum threshold, by looking at descriptions whose confidence level as calculated in Eq. 8 are higher than it. (c) The percentage of descriptions (methods 1-4) an attribute was used in for a select set of attributes. The attributes are: (1) Gender (2) White (3) Black hair (4) Eyeglasses (5) Smiling (6) Chubby (7) Fully visible forehead (8) Eyes open (9) Teeth not visible (10) Beard

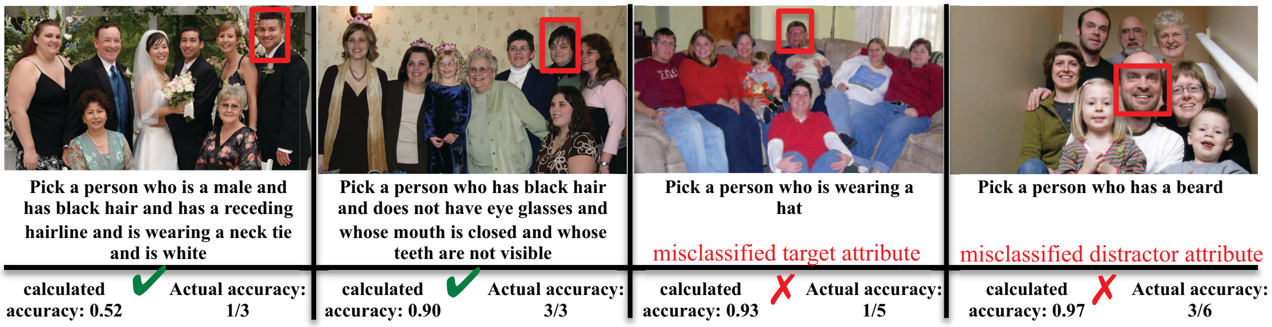

Pick a person who is a male and has black hair and has a receding hairline and is wearing a neck tie and is white
calculated accuracy: 0.52
✓
Actual accuracy: 1/3
Pick a person who has black hair and does not have eye glasses and whose mouth is closed and whose teeth are not visible
calculated accuracy: 0.90
✓
Actual accuracy: 3/3
Pick a person who is wearing a hat
misclassified target attribute
calculated accuracy: 0.93
✗
Actual accuracy: 1/5
Pick a person who has a beard
misclassified distractor attribute
calculated accuracy: 0.97
✗
Actual accuracy: 3/6

Figure 6. Examples of our GBM algorithm along with the calculated confidence and the actual accuracy received from AMT. The left two are examples where our algorithm correctly estimates the confidence (approximately). The right two examples are failure cases: A misclassified target attribute (no hat on target) and a misclassified distractor attribute (additional bearded person in the image).

used as reference instead of the target person. Finally, it appears that some people confused left and right. That said, we do observe clear cases in which using neighbors led to better results (See Figure 7b for an example).

It is also interesting to investigate how guesser accuracy changes as we change the confidence threshold (Figure 5b). Since many of the faces in our algorithm did not reach the necessary confidence, the average confidence of the descriptions is 0.6484 which gives us 60% correct human guesses. However, Figure 5b shows that as we increase the minimum confidence, and look only at the descriptions which are above it we can achieve much higher human guessing accuracy. This validates the meaningfulness of our confidence score. In addition, this shows another strength of using GBM since the Full_greedy approach does not present a simple way of calculating this confidence.

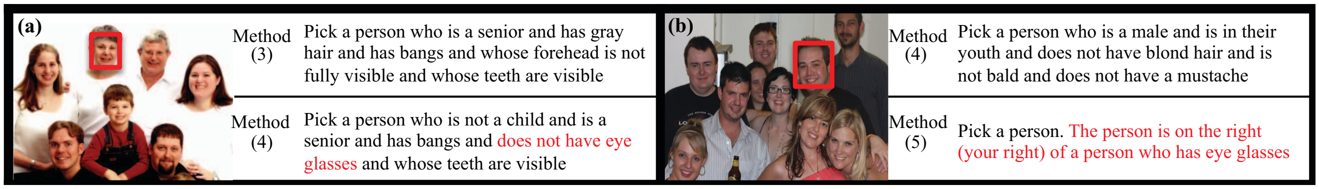
4.2. Human Describers

We also compare our results using computer descriptions with that of a human describer. In an additional AMT job, workers select attribute-value pairs that best refer to the target person. We reduce the number of attributes to 20 (to simplify the task), and present three radio buttons for each attribute: *not needed*, *yes*, *no*. This is exactly analogous to the computer algorithm and therefore the results are easily comparable. Workers select the fewest attributes that separate the target person from the rest of the group (just as our algorithm does). To encourage workers, we promise a bonus to those whose descriptions give the best guessing probability. We collected 1000 descriptions from 100 separate workers.

Once we have collected all the descriptions given by the workers we create a new guessing task as described in Sec. 4.1. We compare the descriptions created by humans to descriptions created by GBM using the same 20 attributes as given to the user. For this comparison we only use descriptions whose confidence is above 0.7. The descriptions created from the human selection are presented to the guesser in the exact the same format as the computer's. The guesser is never informed of the source of the descriptions (human or computer).

Accuracies from the human and computer descriptions are 76% and 77% respectively. This result validates our model, matching human performance when it attains high confidence of guesser success.

Other interesting observations include that humans tend to use gender much more often than any other attribute (about 70% of the descriptions included gender), while this



Method (3) Pick a person who is a senior and has gray hair and has bangs and whose forehead is not fully visible and whose teeth are visible

Method (4) Pick a person who is not a child and is a senior and has bangs and does not have eye glasses and whose teeth are visible

Method (4) Pick a person who is a male and is in their youth and does not have blond hair and is not bald and does not have a mustache

Method (5) Pick a person. The person is on the right (your right) of a person who has eye glasses

Figure 7. Examples of our algorithms output using the methods described in Sec. 4.1. (a) Since algorithm (3) calculates the probability one attribute at a time, all of the attributes it describes could be true for both seniors. However, once narrowed down to the seniors it is enough to say: "does not have glasses" as done by algorithm (4). (b) In this photo finding attributes which refer strictly to the target person without using neighbors (4) is hard. But by using the person with the glasses as a landmark, we can quickly refer to the correct person.

is not true for the computer algorithm. Even in situations where gender is not necessarily needed, humans still tend to mention it. In addition, humans tend to choose more positive attributes rather than negative ones. In fact, of the 19 attributes (excluding gender since there is no negative for this attribute) 18 were mentioned more often positive than negative. In contrast, for 6 of the 19 attributes, our algorithm mentions the negative attributes more often.

5. Conclusion

We have introduced a new approach for solving the novel task of producing a referring expression for a person in an image. We compute a confidence score for each description, based on a novel, efficient method for calculating the score. Finally, we demonstrate the effectiveness of our attribute selection algorithm, comparable even to constrained human-made descriptions.

We believe there are many exciting future directions for this work. First, more can be learned from our human describers and guessers. Our guesser model still does not completely mimic a human because it does not consider factors such as saliency or relative attributes. By examining the human descriptions and guesses, we may learn a better model for the human guesser and redesign our algorithm for referring expression generation.

In addition, this work can be extended to consider back-and-forth conversations between humans and computers. That is, if the referring expression isn't clear, what questions can the guesser ask to clarify her understanding? This might involve answering a user's clarifying question, or providing feedback to a user who guessed incorrectly.

Finally, we believe our framework is an important component for any image description algorithm, though challenges remain dealing with integrate more general image descriptions (e.g., not just referring expressions).

Acknowledgments Work supported in part by NSF DMS-0808864.

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.

[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] R. Dale. Cooking up referring expressions. In *ACL*. Association for Computational Linguistics, 1989.

[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.

[5] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[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.

[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.

[8] A. Gallagher and T. Chen. Finding rows of people in group images. In *ICME*, 2009.

[9] A. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009.

[10] P. Grice. Logic and conversation. *Syntax and Semantics*, 3:43–58, 1975.

[11] A. Gupta, Y. Verma, and C. Jawahar. Choosing linguistics over vision to describe images. In *AAAI*, 2012.

[12] H. Horacek. Generating referential descriptions under conditions of uncertainty. In *ENLG*, 2005.

[13] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012.

[14] E. Krahmer, S. Erk, and A. Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72, 2003.

[15] 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.

[16] 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.

[17] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[18] 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.

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

[20] A. Parkash and D. Parikh. Attributes for classifier feedback. In *ECCV*, 2012.

[21] A. Sadovnik, Y. Chiu, N. Snavely, S. Edelman, and T. Chen. Image description with a goal: Building efficient discriminating expressions for images. In *CVPR*, 2012.

[22] W. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012.

[23] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD*, 2002.