Computer Speech and Language (2002) **16**, 353–385
doi:10.1016/S0885-2308(02)00024-4
Available online at http://www.idealibrary.com on IDEAL®

Learning visually grounded words and syntax for a scene description task

Deb K. Roy[†]

The Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA, U.S.A.

Abstract

A spoken language generation system has been developed that learns to describe objects in computer-generated visual scenes. The system is trained by a 'show-and-tell' procedure in which visual scenes are paired with natural language descriptions. Learning algorithms acquire probabilistic structures which encode the visual semantics of phrase structure, word classes, and individual words. Using these structures, a planning algorithm integrates syntactic, semantic, and contextual constraints to generate natural and unambiguous descriptions of objects in novel scenes. The system generates syntactically well-formed compound adjective noun phrases, as well as relative spatial clauses. The acquired linguistic structures generalize from training data, enabling the production of novel word sequences which were never observed during training. The output of the generation system is synthesized using word-based concatenative synthesis drawing from the original training speech corpus. In evaluations of semantic comprehension by human judges, the performance of automatically generated spoken descriptions was comparable to human-generated descriptions. This work is motivated by our long-term goal of developing spoken language processing systems which grounds semantics in machine perception and action.

1. Introduction

A growing number of applications require the translation of perceptual or sensory data into natural language descriptions. Automatic sports commentators in video games map spatial relations and dynamic events of virtual players into speech. Car navigation systems generate spoken directions based on map routes and geographical position data. We envision assistive aids which translate visual information into speech for the visually impaired. Most current approaches to this class of language generation problems rely on manually created rules which encode domain specific knowledge. These rules are used for all aspects of the generation process including, for example, lexical selection and sentence frame selection. In this paper, we develop a learning-based

[†] E-mail: dkroy@media.mit.edu

0885-2308/02/$35

approach for creating spoken language generation systems. Our ultimate goal is to develop trainable systems that can learn domain specific rules of language generation from examples produced directly by domain experts. We present an implemented system called DESCRIBER which represents a first step in this direction.

We consider the problem of generating spoken descriptions from visual scenes to be a form of *language grounding* (Roy, 2000; Roy, 2000/2001; Roy, in press; Roy & Pentland, 2002). Grounding refers to the process of connecting language to referents in the language user's environment.[1] In contrast to methods which rely on symbolic representations of semantics, grounded representations bind words (and sequences of words) directly to non-symbolic perceptual features.[2] Crucially, bottom-up sub-symbolic structures must be available to influence symbolic processing (Roy, 2000/2001). All symbolic representations are ultimately encoded in terms of representations of the machine's environment which are available to the machine directly through its perceptual system.

We present a grounded system, DESCRIBER, that learns to generate contextualized spoken descriptions of objects in visual scenes. Input to DESCRIBER consists of visual scenes paired with naturally spoken descriptions and their transcriptions. A set of statistical learning algorithms extract syntactic and semantic structures which link spoken utterances to visual scenes. These acquired structures are used by a generation algorithm to produce spoken descriptions of novel visual scenes. Concatenative synthesis is used to convert output of the generation subsystem into speech. In evaluations of semantic comprehension by human judges, the performance of automatically generated spoken descriptions is found to be comparable to human-generated descriptions.

1.1. Related work

The problem of generating referring expressions in text and multimedia environments has been addressed in many previous computational systems including the work of Dale (1992), Dale and Reiter (1995) and André and Rist (1995). Dale's system, EPICURE, addresses the problem of generating anaphoric referring expressions. A planner maps communicative goals to surface text making use of models of the discourse, the hearer, and the world state. Dale and Reiter examined computationally efficient methods for generating expressions which use the minimal set of attributes to distinguish an intended referent from a set of competitor referents. André and Rist designed a generation system which combines text and image output to refer to objects. These and most other previous generation systems may be contrasted with our work in three significant ways. First, our emphasis is on learning all necessary linguistic structures from training data. Second, we take the notion of grounding semantics in sub-symbolic representations to be a critical aspect of linking natural language to visual scenes. Third, we limit the scope of our work to generating referring expressions based solely on information available in static visual scenes. Thus, discourse history is not used by our system.

[1] Grounding symbols in the physical world may be argued to lay the foundation for representing a large range of concepts at many levels of abstraction through analogical and metaphorical reasoning (Lakoff & Johnson, 1980; Harnad, 1990; Barsalou, 1999).

[2] Semantics may also be grounded in terms of actions which the language user performs on its environment. This aspect of grounding is not considered in this paper.

The Visual Translator system (VITRA) (Herzog & Wazinski, 1994) is a natural language generation system which is grounded directly in perceptual input. VITRA generates natural language descriptions of dynamic scenes from multiple domains including automobile traffic and soccer games. Semantic representations are extracted from video image sequences. Detailed domain knowledge is used to categorize spatial relations between objects and dynamic events. Higher level propositions are formed from these representations which are mapped to natural language using a rule-based text planner. An 'imaginary listener' predicts what the listener is most likely to understand from a proposed description. Any disagreements between the predicted message and the intended message are fed back into the text planner until expected ambiguities are minimized. In contrast to our work, VITRA is not designed as a learning system. Thus porting it to a new domain would presumably be a arduous and labor intensive task.

Jordan and Walker (2000) used machine learning to train a system which generates nominal descriptions of objects. Each nominal expression consists of up to four attributes. The learning system was trained to automatically select which subset of attributes to use in a referring expression (i.e. a choice from 1 of 16 possible combinations of four attributes). The decision process is based on a set of dialog context features (for example, what is assumed to be known by the hearer, attributes used recent references to the same object, etc.). The learning algorithm acquires an optimal set of rules by which to map context features into attribute selections which are, in turn, used to generate referring expressions. In comparison to DESCRIBER, Jordan and Walker's approach uses a much richer set of features which encode dialog context. DESCRIBER does not encode any history of interaction and relies solely on features extracted from a static visual scene. The scope of what is learned by DESCRIBER, however, is significantly broader. In addition to attribute selection, syntactic structures and the visual semantics of words are also acquired by DESCRIBER.

Learning grounded representation of spatial terms has been studied by Regier (1996). He designed a set of psychologically motivated perceptual features that underlie spatial concepts across a wide range of languages. For example, to represent static spatial relations such as *above* and *beside*, a pair of relative angles which take into account the center of mass and points of closest proximity between objects was proposed. We employed these features in the system described in this paper. Regier's system acquired spatial terms using connectionist learning methods. Input consisted of synthetic images of pairs of objects and their singleword labels. Regier's work demonstrates the importance of choosing perceptual features carefully to ensure efficient concept acquisition. In related work, Siskind (2001) has proposed the use of visual primitives which encode notions of support, contact, and attachment to ground the semantics of events for verb learning.

In our own previous work (Roy, 2000, Roy, in press), we have modeled the early stages of word acquisition from sensor-grounded speech and visual signals. We demonstrated the utility of learning algorithms based on cross-modal mutual information in discovering words and their visual associations from untranscribed speech paired with images of three-dimensional everyday objects. An implemented system was able to learn object names from a corpus of spontaneous infant-directed speech. A focus of this work was the discovery and segmentation of word-like acoustic units from spontaneous speech driven by cross-modal analysis. An acquired lexicon of visually grounded words served as the basis for a small vocabulary speech understanding and generation system. The system was able to process single and two-word phrases which referred to the color and shape of objects. The language processing system was integrated into an interactive robot. A person was able to issue verbal commands ("red ball") and the robot would actively search for the best matching referent. The system was also able to verbally describe novel objects using two-word phrases. The system presented in this paper extends our prior work in that it addresses the problem syntactic structure acquisition within a grounded learning framework.

1.2. The learning problems

In this paper, we consider learning problems in which each training example is comprised of (1) a natural language word sequence and (2) a vector of real-valued features which represents the semantics of the word sequence. We assume no prior knowledge about lexical semantics, word classes, or syntactic structures.

A basic problem is to establish the semantics of individual words. To bootstrap the acquisition of word associations, utterances are treated as "bags of words." Each word in an utterance may potentially be a label for any subset of co-occurring visual features. Consider the situation in which we measure three features of an object as potential grounding for adjective terms: height, area, and brightness. A person looks at an object and says, "That is a big red apple." In the bag of words model, any word in the sentence (including "that" or "a") might refer to any subset of the measured features. Possibilities include associations that we would like the learner to make such as "big" with area, as well as countless associations which are undesirable such as "red" or "a" with height and brightness. Thus one problem facing the language learner is feature selection: choosing the subset of potential features which should be bound to a word. Once feature assignments have been made, statistical learning methods can be used to train classifiers which map words to ranges of values within those features. For example "dark" might select only the brightness feature and prefer small values of that feature.

A second problem is to cluster words into word classes based on semantic and syntactic constraints. We assume that word classes are a necessary first step in acquiring rules of word order. For example, before a language learner can learn the English rule that adjectives precede nouns, some primitive notion of adjective and noun word classes presumably needs to be in place. Word classes might be derived strictly from distributional analysis of word co-occurrences. Alternatively, semantic associations might be used to group words. In this paper, we present a hybrid method which combines distributional and semantic cues.

A third problem is learning word order. We address the problems of learning adjective ordering ("the large blue square" vs. "the blue large square") and phrase ordering for generating relative spatial clauses. In the latter, the semantics of phrase order needs to be learned (i.e. the difference in meaning between "the ball next to the block" vs. "the block next to the ball"). A statistical bigram language model is learned in terms of the acquired word classes which is used to generate compound word descriptions of objects such as "the thin dark green rectangle." Statistical bigrams of phrases are employed to model syntactic structures necessary for generating relative spatial clauses. The semantic implications of phrase order are captured in the grounding of spatial lexical items in terms of relative spatial visual features.

Once the problems outlined above have been addressed, the system has at its disposal a grounded language model which enables it to map novel visual scenes into natural language descriptions. The language generation problem is treated as a search problem in a probabilistic framework in which syntactic, semantic, and contextual constraints are integrated.

1.3. Outline

This paper begins by outlining the experimental task and training corpus which we have created as a development test bed. Algorithms for learning to generate natural language expressions which refer to single objects are presented. A second set of algorithms are then presented which enables the system to generate expressions that include relative spatial clauses. These clauses help listeners disambiguate between similar objects in the visual scene. Finally, an evaluation of the system is presented in which the system's performance in semantic understandability is compared with the original human produced training corpus.

2. The visual description task

The experiments reported in this paper are based on a *rectangle description task*. A program was created to generate images consisting of a set of 10 colored rectangles set on a black background. A typical image generated by this program is shown in Figure 1. The width, height, position, and RGB color of each rectangle is randomly

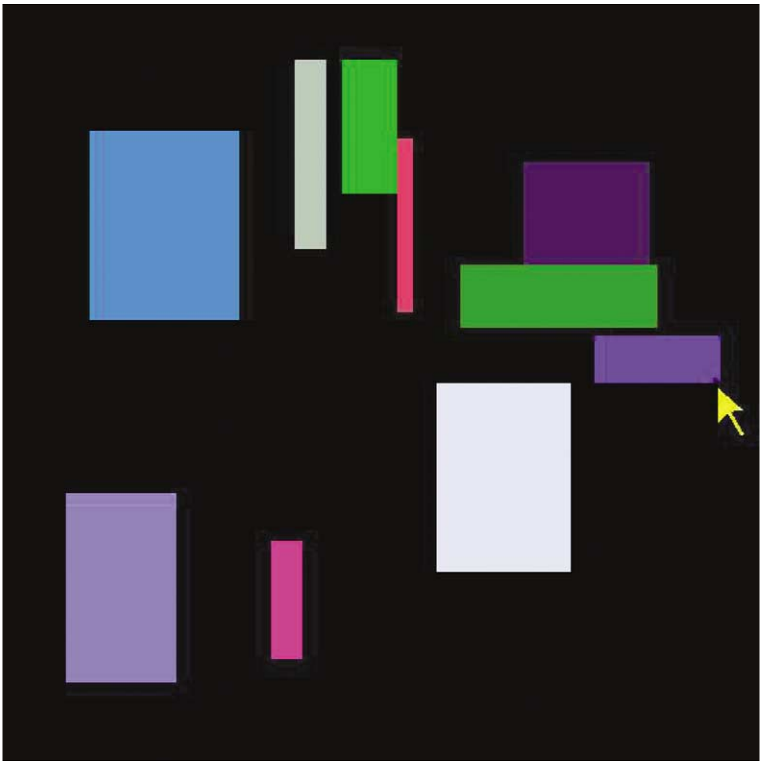


Figure 1. A typical image from the rectangle task. Each synthetic image consists of 10 rectangles, each of random height, width, color, and non-overlapping position.

generated. The placement of rectangles is constrained such that they never overlap although they may touch. During data collection (described below), each image is augmented with an indicator arrow which selects one of the 10 rectangles as the *target object*. For example, in Figure 1 the blue rectangle on the right is the target.

The description task consists of generating phrases which best describe target objects. The generation process must be context sensitive since the best choice of words will often depend on the other objects in the scene. Descriptions are evaluated for semantic understandability by measuring how reliably human listeners select the intended target object from the same scene based on the provided verbal description.

This task was chosen as a manageable starting point for our experiments. The variation of objects is limited to shape, color, size, and position. The syntactic structure required to generate descriptive phrases is relatively simple and well modeled by statistical n-grams. By using computer-generated images, visual feature extraction is greatly simplified (when compared to using camera images). Nonetheless, we found that the challenges raised in this task were substantive and lead to useful new algorithms. Results from this task will form the basis for future explorations of more complex language learning tasks in richer contexts.

3. Visual features

Lexical semantics are grounded in terms of visual features. Table I lists the set of eight visual features that are generated by the image synthesis program to represent each object in an image. We refer back to these features in the remainder of the paper using the names listed in the left column of this table.

These features were selected with the language learning task in mind. For example, we would expect color terms to be grounded in some combination of the r, g, and b features. Spatial terms such as "leftmost" and "highest" should be grounded in the x and y features. Other words such as "bright" or "thin" are less obvious. Features are normalized so that each feature has zero mean and unit variance.

4. Data collection and preparation

We collected a speech corpus from a male speaker (an undergraduate student unfamiliar with the project). He was instructed to speak naturally and describe target objects from images displayed on a computer screen. He was asked to produce descriptions such that a listener could later select the same target from the identical scene with the target unmarked.

TABLE I. Visual features extracted from objects

Name	Description
r	Red component of RGB color
g	Green component of RGB color
b	Blue component of RGB color
hw_ratio	Height to width ratio
area	Surface area
x	X position of upper left corner
y	Y position of upper left corner
mm_ratio	Ratio of maximum dimension to minimum dimension

TABLE II. Typical utterances in the rectangle task corpus

Type	Utterance
Simple	The pink square
Simple	The light blue square
Simple	The biggest grey rectangle
Simple	The large off white rectangle
Simple	The long flat purple rectangle
Simple	The brightest green rectangle
Complex	The narrow purple rectangle below and to the right of the blue square
Complex	The green rectangle below the peach rectangle
Complex	The purple rectangle to the left of the pink square
Complex	The orange rectangle above the blue rectangle
Complex	The yellow rectangle to the left of the large green square
Complex	The vertical rectangle directly below the smallest blue rectangle

Simple utterances contain reference to exactly one object. Complex utterances refer to multiple objects.

A data collection program was written which displays images and records spoken responses. In preparation for data collection, a set of 3000 images were generated off-line, each with a randomly selected target object. The speaker wore a noise-canceling headset microphone. The presentation program displayed each image and recorded the speaker's spoken response. An on-line speech end-point detection algorithm based on Hidden Markov models of speech and silence (Yoder, 2001) was used to segment incoming speech into utterances. Each segmented utterance was saved as a separate speech file. Each file was automatically tagged with the identity of the image and target object on display at the time.

The speaker participated in two 90-min recording sessions resulting in 518 utterances.[3] Each spoken utterance was manually transcribed at the word level. We divided training utterances into two types: *simple utterances* and *complex utterances*. Simple utterances contain reference to exactly one object whereas complex utterances make reference to two or more objects. Classification of utterances was based on text keyword spotting. Any transcript containing multiple instances of the words "rectangle" or "square" was classified as complex and the remainder as simple. Table II lists some representative utterances of each type from the corpus. Out of the total 518 utterances, 326 are simple and 192 complex. The mean utterance length in the corpus is 5.8 words. The mean utterance length of simple utterances is 4.0 words.

An initial histogram analysis of the corpus indicated insufficient exemplars of some color and spatial terms. A second speaker was asked to provide an additional set of descriptions focused on color and spatial terms. This speaker was instructed to produce only simple utterances. New random images where generated for this collection. An additional 157 simple utterances were collected in a single recording session. Put together with the data from the first speaker, the training corpus consisted of 675 utterances (483 simple, 192 complex).

A complication in learning from this data is that complex utterances contain reference to multiple objects. In a bag of words model, any word must be considered a label for any co-occurring observation. To simplify the problem, we truncated each transcription of a complex utterance after the first instance of either "rectangle" or

[3] Although 3000 images were generated, only 518 of them were used in the data collection. The 518 images were randomly chosen from the set of 3000.

"square." The truncated transcripts were used for the first stages of learning (described in Section 5). Learning from whole (untruncated) utterances is addressed in Section 7. The truncation procedure is based on knowledge of the task at hand (i.e. the fact that the first object phrase most likely will refer to the target object) and will not necessarily generalize to other situations. In the future, we plan to develop methods to avoid this simplification.

Appendix A contains histograms of word occurrences in the original and truncated training corpora. The full training corpus contains one or more instances of 83 unique words (i.e. 83 token types). The truncated corpus draws from 70 token types.

The manual speech transcription process is the most labor intensive aspect of the training process. In the future, we will explore the use of speech recognition to automate this step, but initially we preferred to work with error free transcripts.

5. Learning grounded language models for objects

This section describes the set of algorithms which have been developed for acquiring the structures necessary to produce simple utterances as defined in the previous section. The order of presentation of algorithms corresponds to the stages of processing in the system. Section 6 describes how these structures are employed in generating visually grounded object description phrases.

5.1. Word class formation

The first stage of learning is to cluster words into classes. These classes serve two roles. First, they are used to determine which visual features are associated with a word. All members of a word class are required to be grounded in the same set of features. Second, word classes form the basis for learning a class-based bigram language model. Class-based bigrams enable the system to generalize knowledge from training data to novel word combinations.

Ideally, words which are both semantically and syntactically similar should be clustered together. For example, color terms should be clustered and separated from size and spatial terms. If they were all treated as part of one word class, then learning to differentiate "large blue ball" vs. "blue large ball" would be impossible. Although such knowledge might be preprogrammed, our goal is to develop an extensible system which can form word classes in the absence of manually encoded structure.

We investigated three approaches to forming word classes. The first relies only on the distributional patterns of words in the training corpus and ignores visual information. The second approach searches for associations between words and visual features and then groups words which have similar feature associations. The third approach, which we found to be most effective, is a hybrid method which combines the first two approaches.

5.1.1. Distributional clustering

The distributional method rests on a basic assumption: words belonging to the same class will be used in mutual exclusion. From this assumption it follows that two words which co-occur in the same utterance are likely to belong to different word classes. The utterance, "the large blue square" lends evidence against placing "the" and "large," or "large" and "square," or any other word pair in the same class. This assumption is similar to the mutual exclusion bias that has been proposed as a mechanism used by children in language acquisition (Markman, 1991). Young children initially resist learning two labels for the same concept such as "poodle" and "dog." This bias leads to efficient learning from limited examples.

We will denote a corpus of M utterances by $U = \{u_1, u_2, \ldots, u_M\}$. The vocabulary of the corpus (i.e. the set of unique token types) is denoted by $W = \{w_1, w_2, \ldots, w_V\}$, where V is the vocabulary size and w_i is a word in the vocabulary. A co-occurrence indicator variable is defined by

$$\pi(u, w_i, w_j) = \begin{cases} 1 & \text{if } w_i \text{ and } w_j \text{ occur in } u, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

i.e. $\pi(u, w_i, w_j)$ detects when both words w_i and w_j occur in the utterance u. Based on π, we obtain $V \times V$ co-occurrence matrix R, the elements of which are computed by

$$R(w_i, w_j) = \sum_{u \in U} \pi(u, w_i, w_j), \tag{2}$$

$R(w_i, w_j)$ is the count of the number of times words w_i and w_j co-occur in an utterance, accumulated across all M utterances in the corpus.

Our goal is to partition the vocabulary into word classes such that words within a class co-occur infrequently with other words in the same class. A clustering procedure is used to partition the vocabulary W into K disjoint classes C_j each with N_j words. The kth word in word class j is $C_j(k)$. Since the word classes partition the original vocabulary W, $C_j(k) \in W$. Before specifying the clustering algorithm, we first define a distortion metric between two word classes as

$$d_{\mathrm{d}}(C_i, C_j) = \frac{\sum_{k=1}^{N_i} \sum_{l=1}^{N_j} R(C_i(k), C_j(l))}{\sum_{k=1}^{N_i} \langle C_i(k) \rangle + \sum_{k=1}^{N_j} \langle C_j(k) \rangle}, \tag{3}$$

where $\langle C_i(k) \rangle$ is the count of the number of times word k from class i occurred in the training corpus. The subscript d in $d_{\mathrm{d}}(\,)$ reminds us that this distortion metric is based on distributional cues. The numerator of Equation (3) accumulates all co-occurrences of each pair of words drawn from class i and j. The denominator normalizes this sum by the total number of occurrences of both word classes in the training corpus.

A greedy clustering algorithm is used to iteratively merge clusters with smallest inter-cluster distortion until a stopping criterion is met. The algorithm consists of five steps:

1. Begin with $K = V$ clusters, each initialized with one word from W.
2. Find C_i and C_j such that $d_{\mathrm{d}}(C_i, C_j)$ is minimized, $1 \leqslant i, j \leqslant K$.
3. If $d_{\mathrm{d}}(C_i, C_j) > T$ then stop, for some stopping threshold T.
4. Merge elements of C_i and C_j.
5. Go to Step 2.

The value of the stopping threshold T determines the number of word classes produced. We have not developed an automatic method to determine the optimal value of T. One possibility is to adapt its value in a reinforcement learning framework. Currently, however, this value is set manually.

TABLE III. The 10 word classes created after 22 merges based on distributional analysis of within-utterance word co-occurrences

Word class	Class members
0	the
1	light grey white dark bright leftmost salmon highest
2	pink blue yellow green purple red brown orange colored
3	horizontal vertical square largest
4	rectangle
5	small large thin smallest lowest
6	tall
7	olive
8	off
9	rightmost

The clustering algorithm was applied to the training corpus. To avoid estimation problems due to small sample sizes, all words occurring less than five times in the corpus were removed from the vocabulary and from all further processing. After removal of infrequent words, the experimental corpus[4] consisted of 32 unique word types. Table III lists the 10 word classes formed using distributional analysis after 22 merges. An examination of the word classes reveals that the mutual exclusion bias approximately separates color terms (Class 2), shape descriptors (Class 3), and size descriptors (Class 5). However, errors are also evident in the classes. Some color terms (grey, salmon) are included with non-color terms in Class 1 and spatial terms (leftmost, rightmost, highest, lowest) have not been clustered.

5.1.2. Clustering based on semantic feature associations

We investigated a second method of clustering which ignores co-occurrence patterns of words within utterances and instead focuses on semantic associations between words and visual referents. The goal of this approach is to cluster words which are grounded in similar sets of visual features. In principle, many of the errors introduced by distributional analysis could be resolved by factoring in semantic constraints.

One problem in establishing semantic associations of words is that natural language does not provide exhaustive labels of all referents in a scene. Consider an image in which only one object is red. If asked to describe that object (relative to the others), a person might say something to the effect of "the red one." Various other possible descriptions of the object such as its size, its location, etc., are absent from the description. When learning from natural language descriptions, we cannot assume that the absence of a label indicates the absence of the corresponding property. If the person did not use the word "large," we are unable to conclude that the object is not large. The problem of lack of negative training examples is well known in the context of grammar acquisition in children (cf. Braine, 1971) and arises also in the case of lexical acquisition.

In our approach, each visual feature is treated as a random variable which is modeled with a univariate Gaussian distribution. We begin by quantifying the effect of the presence of each word on the distribution of each feature. A semantic distortion metric which operates on word pairs will then be defined in terms of these individual

[4] Recall that for all training described in this section, we used the corpus with utterances truncated after the first instance of the keywords "rectangle" or "square."

feature effects. Finally, this distortion metric will be incorporated into the word clustering algorithm.

Recall that each utterance u_i in the training corpus is paired with a target object. From each object, F visual features are extracted ($F = 8$ in our current experiments). The F-dimensional feature vector extracted from the object paired with utterance u_i is referred to as x_i. We refer to feature j of x_i as $x_i(j)$. To model the effect of word w_n on the distribution of feature $x(j)$, only the observations which occur in the presence of an utterance containing w_n are used to obtain the unbiased estimates of the Gaussian parameters of a *word-conditional model*:

$$\mu_{j|w_n} = \frac{\sum\limits_{i,w_n \in u_i} x_i(j)}{\sum\limits_{i,w_n \in u_i} 1}, \tag{4}$$

$$\sigma_{j|w_n} = \frac{\sum\limits_{i,w_n \in u_i} (x_i(j) - \mu_{j|w_n})^2}{\left(\sum\limits_{i,w_n \in u_i} 1\right) - 1}. \tag{5}$$

The summations are over all utterances which contain the word w_n. Note that the denominator in Equation (4) may be smaller that $\langle w_n \rangle$ since the former does not count multiple instances of a word within an utternance while $\langle w_n \rangle$ does. The two terms would be equal only if w_n never occurs more than once within the same utterance.

The remaining observations which did not co-occur with w_n are used to estimate the parameters of a background model:

$$\mu_{j|\overline{w_n}} = \frac{\sum\limits_{i,w_n \notin u_i} x_i(j)}{\sum\limits_{i,w_n \notin u_i} 1}, \tag{6}$$

$$\sigma_{j|\overline{w_n}} = \frac{\sum\limits_{i,w_n \notin u_i} (x_i(j) - \mu_{j|\overline{w_n}})^2}{\left(\sum\limits_{i,w_n \notin u_i} 1\right) - 1}. \tag{7}$$

We wish to quantify the distortion between the word-conditioned and background distributions of each feature as a measure of the degree of association between the word and the feature. The Kullback–Leibler (KL) divergence (Cover & Thomas, 1991) provides an asymmetric measure of dissimilarity between two probability distribution functions p and q and is given by

$$\mathrm{KL}(p\|q) = \int p(x) \ln \frac{p(x)}{q(x)}. \tag{8}$$

An extension of the KL divergence which provides a symmetric distance between distributions is

$$\mathrm{KL}_2(p\|q) = \mathrm{KL}(p\|q) + \mathrm{KL}(q\|p). \tag{9}$$

We refer to Equation (9) as the *symmetrized KL distance*. The symmetrized KL distance is used to compare the unconditioned and word-conditioned distribution of a feature:

$$\mathrm{KL}_2(p(x_j|\overline{w_i})\|p(x_j|w_i)) = \frac{1}{2}\left(\frac{\sigma^2_{j|\overline{w_i}}}{\sigma^2_{j|w_i}} + \frac{\sigma^2_{j|w_i}}{\sigma^2_{j|\overline{w_i}}} - 2\right) + \frac{1}{2}(\mu_{j|\overline{w_i}} - \mu_{j|w_i})^2\left(\frac{1}{\sigma^2_{j|\overline{w_i}}} + \frac{1}{\sigma^2_{j|w_i}}\right). \tag{10}$$

The symmetrized KL distance is always positive (or zero when the distributions are equal) and provides a measure of association between words and individual visual features.

We wish to define a semantic distortion metric which will be used in place of the distributional distortion (Equation (3)) in order to form word classes. For each word w_n we compute a corresponding *semantic association vector* as the collection of feature-wise KL distances

$$s(w_n) = \begin{pmatrix} \mathrm{KL}_2(p(x_1|\overline{\omega_n})\|p(x_1|w_n)) \\ \mathrm{KL}_2(p(x_2|\overline{\omega_n})\|p(x_2|w_n)) \\ \vdots \\ \mathrm{KL}_2(p(x_F|\overline{\omega_n})\|p(x_F|w_n)) \end{pmatrix}. \tag{11}$$

To make comparisons between words, semantic association vectors are linearly scaled such that the largest element in the vector is 1.0 (and the smallest value will be between 0 and 1).

The semantic association vector may be thought of as a "semantic profile" of a word (Gorin developed a similar concept to quantify the semantic associations of words and phrases in relation to discrete actions in a call routing task, Gorin, 1995). Figure 2 shows the scaled semantic association vectors for six words from the training corpus. The word "blue" is associated most strongly with the r and b color features. "Dark" and "light" are both associated with color channels as well, although "light" is not associated with the red feature, but "dark" is. "Rightmost" is associated most strongly, as would be expected, with the x feature (horizontal position). "Square" is associated with both the height over width and min over max features. Surprisingly, "thin" is associated most strongly with area and only weakly with hw_ratio and mm_ratio. We found, in fact, that the speaker in this corpus usually labeled only small objects as "thin."

Based on semantic association vectors, the distortion between two words is defined as the negative of the dot product of the corresponding semantic association vectors, $-[s(w_i)]^{\mathrm{T}}s(w_j)$, where T denotes the transpose operator. The semantic distortion is greatest for word pairs with orthogonal semantic association vectors. The word pair distortion is computed for each pairwise combination of words from two word classes to obtain a semantic distortion between pairs of word classes

$$d_{\mathrm{s}}(C_i, C_j) = \frac{\sum_{k=1}^{N_i}\sum_{l=1}^{N_j} -[s(C_i(k))]^{\mathrm{T}}s(C_j(l))}{N_iN_j}. \tag{12}$$

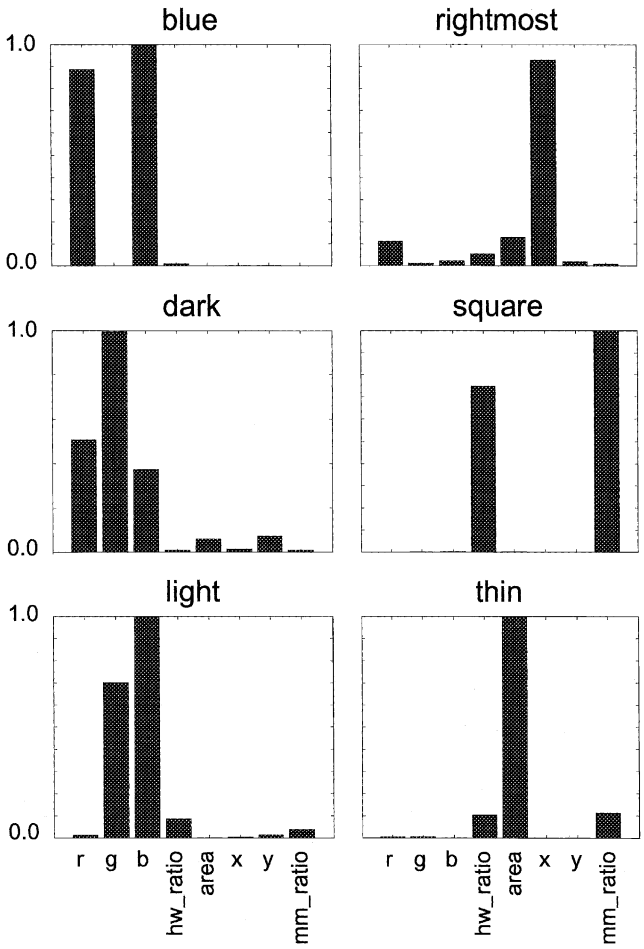


Figure 2. Examples of semantic association vectors for six words.

Table IV. The 10 word classes created after 22 merges based on semantic associations

Word class	Class members
0	the horizontal large vertical rectangle square bright largest
1	light grey red green purple colored dark blue brown
2	pink yellow salmon orange
3	white
4	small thin smallest
5	tall
6	olive
7	leftmost rightmost
8	off
9	lowest highest

The clustering algorithm presented in the previous section was rerun on the corpus with $d_d(\,)$ replaced by $d_s(\,)$. The resulting word classes after 22 merges are listed in Table IV. In contrast to the classes in Table III based on distributional analysis, semantically driven classes display different groupings. For example, “leftmost” and “rightmost” are clustered but kept separate from “lowest” and “highest.” Although semantically related words are now separated, syntactic roles are ignored. Thus “the” and “rectangle” are placed in the same word class as are several shape and size adjectives.

5.1.3. Hybrid clustering

Word co-occurrences and semantic associations are both clearly important cues in forming word classes. Both sources of information can be combined by computing a linear combination of the distortion metrics:

$$d_{\mathrm{ds}}(C_i, C_j) = \alpha d_{\mathrm{d}}(C_i, C_j) + (1 - \alpha) d_{\mathrm{s}}(C_i, C_j). \tag{13}$$

Using $\alpha = 0.95$ and a stopping threshold of $T = -0.6$, we obtained the word classes listed in Table V. The large value of α should not be interpreted as favouring $d_{\mathrm{d}}(\,)$ over $d_{\mathrm{s}}(\,)$. Rather, it compensates for the fact that the range of values of $d_{\mathrm{d}}(\,)$ is smaller than that of $d_{\mathrm{s}}(\,)$. As can be seen by comparison with Tables III and IV, the grouping of words using d_{ds} is significantly effected by both semantic and syntactic constraints. Semantically related words are grouped together, yet syntactically distinct words are kept apart. For example, “square” is grouped with “rectangle” even though semantically, “square” is more similar to “vertical” and “horizontal” (i.e. terms which are associated the ratios of height to width). The words “off,” “tall,” and “olive” are not grouped because their semantic association vectors were not sharp enough to cluster them with other words. This was due to insufficient consistent training data for those words.

The word classes listed in Table V were used in the final generation system.

5.2. Feature selection

Feature selection proceeds by first assigning features to each word on an individual basis. The features of a word class are then defined as the conjunction of all features selected for the members of that class.

Features for an individual word are selected to maximize the symmetrized KL distance between the word conditional distribution and the background unconditioned distribution. In the previous section, only one feature was considered at a time. Now,

TABLE V. The 11 word classes created after 21 merges based on a linear combination of the distributional and semantic association distortion metrics

Word class	Class members
0	the
1	light white dark
2	pink yellow salmon orange grey red green purple colored blue brown
3	horizontal vertical bright
4	rectangle square
5	small thin large largest smallest
6	tall
7	olive
8	leftmost rightmost
9	off
10	lowest highest

we consider multivariate distributions over multiple features. The multivariate extension of the symmetrized KL distance is given by Therrien (1989):

$$\begin{aligned}\mathrm{KL}_2(p_1(x)\|p_2(x)) = &\frac{1}{2}\mathrm{tr}\,(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I)\\ &+\frac{1}{2}(\mu_1 - \mu_2)^{\mathrm{T}}(\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2),\end{aligned} \tag{14}$$

where Σ is the full covariance matrix and μ is now a mean vector. Feature selection is achieved using a greedy algorithm. It starts by selecting the single feature which leads to the highest symmetrized KL distance between conditioned and unconditioned distributions according to Equation (10). Next, a search iteratively finds the next best feature which maximally increases Equation (14). Each KL distance is normalized by the number of selected features (i.e. the number of dimensions). After each feature is added, the increase in normalized KL distance is computed. The search stops when no increase is obtainable.

This feature search algorithm leads to interesting behavior for words that could not be reliably associated with visual features (“off,” “olive,” and “tall” in our corpus): the KL distance is maximized when all eight features are selected. This is because no consistently distinct distribution is found along any subset of features. By modeling all features, the data are overfit. Based on this observation, we added a check for words with all features selected and marked these as *ungrounded.* Ungrounded words are words which occur frequently but the semantics of which are unknown. The remaining words (for which features are successfully selected) are referred to as *grounded words.*

Each word class inherits the conjunction (i.e. the inclusive logical-OR) of the features assigned to all of its members. Table VI shows the features which were selected for each word class. For convenience, the members of each class are listed again. By inspection, the assignment of features matches what we would intuitively expect. The choice of adjective partitions is driven jointly by semantic similarity and co-occurrence patterns. The word “bright” was placed in Class 3 with “horizontal” and “vertical” due largely to the influence of co-occurrence counts. “The” is grounded due to the unusually high frequency of this word. The ‘background’ distribution for “the” according to Equations (6) and (7) was estimated with far fewer observations that the word-conditioned model. Due to this imbalance, the feature selection procedure is able to find a stable assignment of features which separates the models.

TABLE VI. Word class feature selection

Word class	Class members	Features
0	the	hw_ratio, mm_ratio
1	light white dark	g,b
2	pink yellow salmon orange grey red green purple colored blue brown	r, g, b
3	horizontal vertical bright	r, hw_ratio, mm_ratio
4	rectangle square	mm_ratio
5	small thin large largest smallest	area, hw_ratio
6	tall	(ungrounded)
7	olive	(ungrounded)
8	leftmost rightmost	x
9	on	(ungrounded)
10	lowest highest	area, y

5.3. Modeling word semantics

For each grounded word, a Gaussian model is estimated using the observations which co-occur with that word. This is achieved using the multivariate form of Equations (4) through (7). The word-conditional model specifies a probability density function (pdf) over the subset of visual features which have been selected for the corresponding word class. For example, the grounding of each member of Cluster 1 is modeled using a two-dimensional Gaussian distribution over the features g and b.

Figure 3 plots the mean and contours of equal probability density for the words in Word Classes 1 and 5. These classes are simpler to visualize since both are assigned two visual features. In Class 1 (left-hand side of figure), we find significant overlap between the distributions associated with "light" and "white" but clear separation between both from "dark." Word class 5 involves distributions over the features area and mm_ratio (the ratio between the larger dimension of a rectangle to the smaller). The shape of the equal probability ellipse of "thin" indicates that the term refers to objects which have high values of mm_ratio (as expected) *and* small areas.

An interesting problem is grounding the semantics of the morpheme 'est'. The distinction of "small" versus "smallest" and "large" versus "largest" is primarily a matter of degree along the area dimension. The actual semantic distinction between these word pairs cannot be represented in DESCRIBER since the concepts of relative ordering and property comparison necessary to ground 'est' are not supported. In principle, it would be possible provide to a basis for grounding 'est' by adding higher order features which compare and sequence visual attributes.

5.4. Class-based statistical bigram model

The final component of the grounded language model necessary to generate noun phrases is a syntactic component which encodes word order constraints. A class-based bigram statistical language model is used for this purpose. Each word in the training corpus is mapped to its corresponding word class label. The probability that class C_i follows C_j is estimated from relative counts:

$$p(C_i|C_j) = \frac{\langle C_j, C_i\rangle}{\langle C_j\rangle}. \tag{15}$$

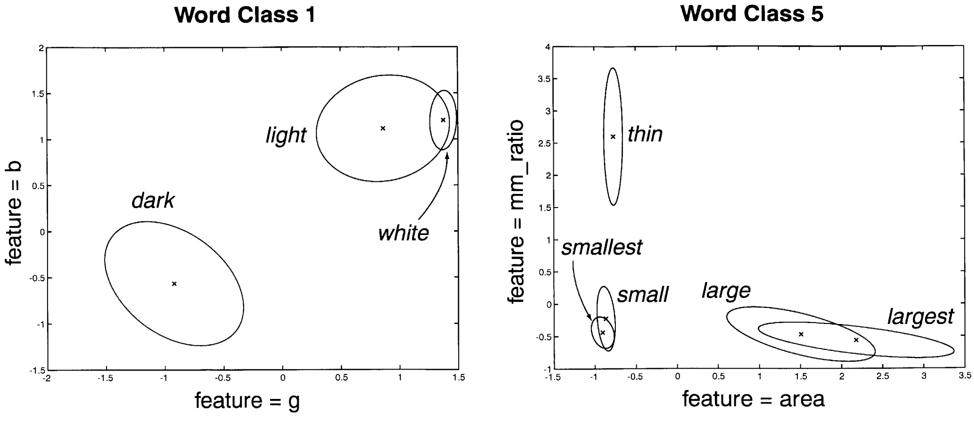


Figure 3. Gaussian distributions associated with words in Word Classes 1 and 5.

The probability of beginning an utterance with a word from C_i is estimated using

$$p(C_i|\text{START}) = \frac{\text{number of times } C_i \text{ at start of utterance}}{M} \tag{16}$$

and similarly the probability of ending an utterance with a word from C_i is estimated using

$$p(\text{END}|C_i) = \frac{\text{number of times } C_i \text{ at end of utterance}}{M}. \tag{17}$$

Turing-Good smoothing (Good, 1953) is optionally used for all three estimates when the numerator is less than 5. As we discuss in later sections, for language generation, smoothing is not always desired. The bigram language model estimated from the training corpus is shown in Figure 4. The transition probabilities are unsmoothed and only transitions with $p > 0.01$ are shown in the figure. Nodes with double outlines indicate the start and end of utterances.

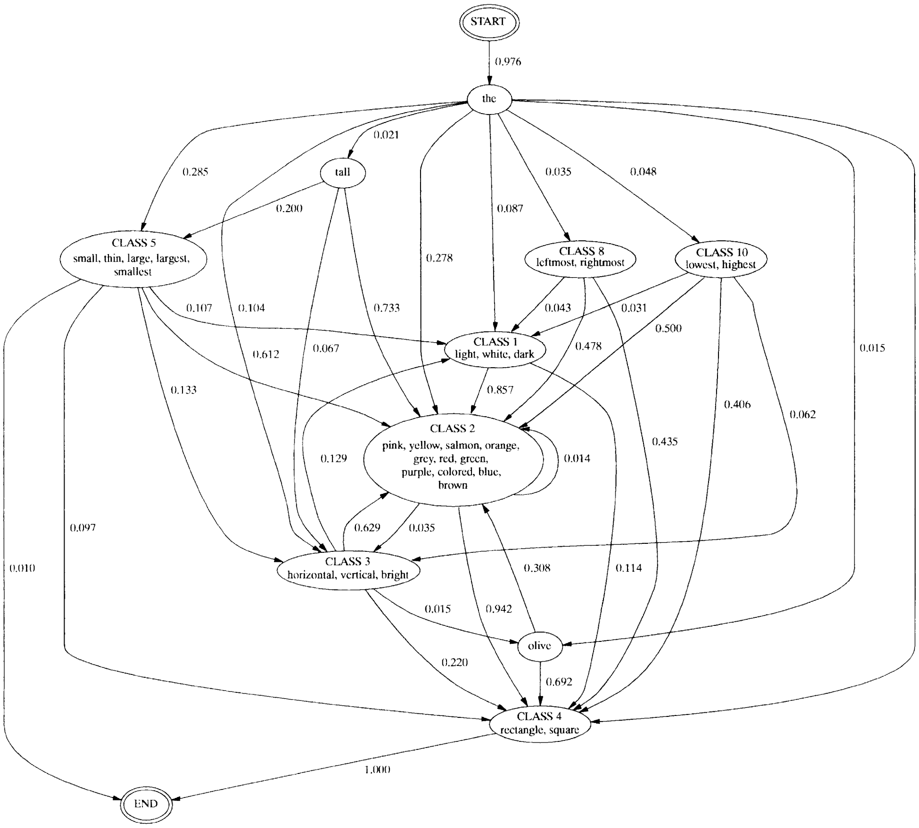


Figure 4. Word-class based statistical bigram for simple utterances.

5.5. Summary

Thus far we have presented a set of algorithms for building a language model from utterances (sequences of words) paired with objects (visual feature vectors). The components of this model are:

- *Word classes*: clusters of words which are grouped according to their distributional (co-occurrence) patterns and semantic associations. Class membership is mutually exclusive (i.e. the same word cannot belong to two classes).
- *Word class features*: a subset of visual features are associated with each word class. The same feature may be linked to multiple word classes.
- *Visually grounded word models*: multivariate Gaussian models associated with each word which model the expected word-conditional distribution of visual features. Word models capture the visual semantics of the word. They only specify a distribution over the features associated with the word's class.
- *Class-based bigrams*: class-based bigram transition probabilities which model word order constraints.

These components provide the basis for generating utterances to describe single objects embedded in visual scenes.

6. Generating spoken language descriptions of objects

We wish to generate natural language phrases of objects which are in some sense optimal given a grounded language model. The generation problem is treated as a constrained search problem. Three types of constraints must be integrated into the search. The first are syntactic constraints since we wish to generate words consistent with natural language syntax. The second constraint is semantic. The semantics of the phrase should describe the features of the target object. A third constraint is context. The phrase must not only describe the features of the object, but should also minimize ambiguity relative to other objects in the scene.

Semantic and contextual constraints are not the same. Consider the scene in Figure 5. The phrase "the large green rectangle" would be a good choice with respect to semantic constraints since the target fits the description well. However, this utterance would be a poor choice when context is factored in since at least one, perhaps two other objects also fit the same description.

We proceed by first describing search constrained only by syntax and then incrementally introduce semantic and contextual constraints.

6.1. Syntactic constraints

We begin by determining the T word utterance which is most likely to be generated using class-based bigrams. We denote an output sequence as $Q = q_1 q_2 \cdots q_T$ where each element q_t is the integer index of a word class, i.e. $1 \leqslant q_t \leqslant K$. The log probability of a word-class sequence is given by

$$\gamma(Q) = \log P(C_{q_1}|\text{START}) + \sum_{t=2}^{T} \log P(C_{q_t}|C_{q_{t-1}}) + \log P(\text{END}|C_{q_T}). \tag{18}$$

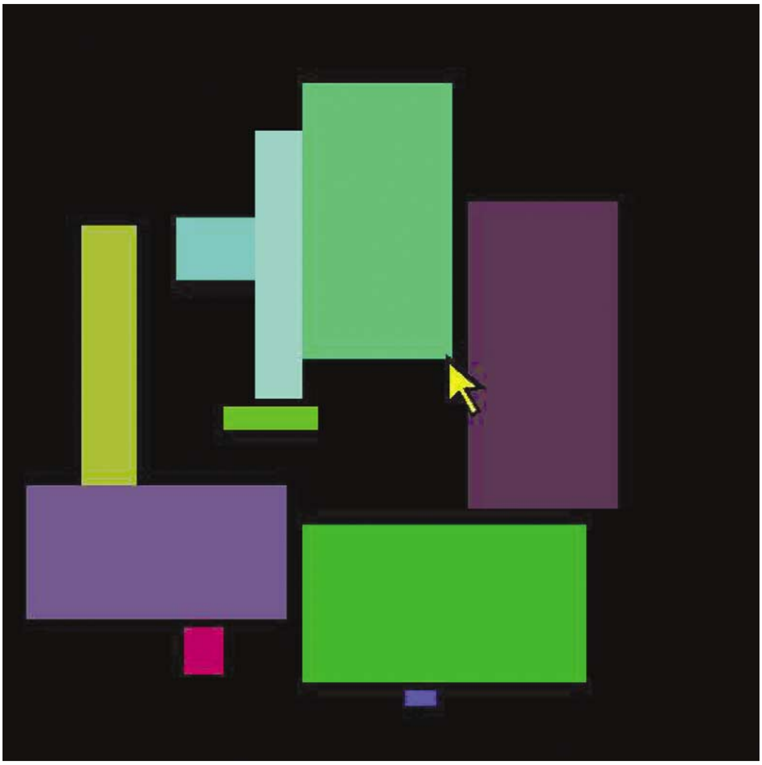


Figure 5. A visual scene with a difficult to describe target object.

The most likely sequence of length T is that which maximizes the total probability of the utterance

$$Q_{\text{best}} = \underset{\text{all } Q}{\arg\max}\ \gamma(Q). \tag{19}$$

Using bigrams estimated from the training corpus, the optimal word class sequence for utterances of increasing length T are

$$\begin{array}{ll}
T = 1 & \langle C_0 : \text{the} \rangle \\
T = 2 & \langle C_0 : \text{the} \rangle \ \langle C_4 : \text{rectangle, square} \rangle \\
T = 3 & \langle C_0 : \text{the} \rangle \ \langle C_2 : \text{pink, yellow}, \ldots \rangle \ \langle C_4 : \text{rectangle, square} \rangle \\
T = 4 & \langle C_0 : \text{the} \rangle \ \langle C_5 : \text{small, thin}, \ldots \rangle \ \langle C_2 : \text{pink, yellow} \rangle \ \langle C_4 : \text{rectangle, square} \rangle \\
T = 5 & \langle C_0 \rangle \ \langle C_5 \rangle \ \langle C_1 \rangle \ \langle C_2 \rangle \ \langle C_4 \rangle \\
T = 6 & \langle C_0 \rangle \ \langle C_5 \rangle \ \langle C_3 \rangle \ \langle C_1 \rangle \ \langle C_2 \rangle \ \langle C_4 \rangle \\
T = 7 & \langle C_0 \rangle \ \langle C_5 \rangle \ \langle C_2 \rangle \ \langle C_3 \rangle \ \langle C_1 \rangle \ \langle C_2 \rangle \ \langle C_4 \rangle \\
T = 8 & \langle C_0 \rangle \ \langle C_5 \rangle \ \langle C_1 \rangle \ \langle C_2 \rangle \ \langle C_3 \rangle \ \langle C_1 \rangle \ \langle C_2 \rangle \ \langle C_4 \rangle \\
\vdots & \vdots
\end{array}$$

An additional constraint is used to avoid repetition of words classes due to loops in the bigram networks. Utterances with word class sequences for $T = 7$ and $T = 8$ shown above are eliminated based on this constraint.

6.2. Mapping word classes to words using visual grounding

Each word class C_i in an output utterance may be mapped to a word by choosing

$$\arg\max_{C_i(j), 1 \leqslant j \leqslant N_i} P(x|C_i(j))P(C_i(j)|C_i), \tag{20}$$

i.e. choose the word $C_i(j)$ from class C_i which maximizes the probability of the target object x. Equation (20) is a standard Bayes classifier using the word-conditional Gaussian models associated with C_i as competing models. The class conditional word probabilities are given by relative word counts

$$P(C_i(j)|C_i) = \frac{\langle C_i(j) \rangle}{\sum_{k=1}^{N_i} \langle C_j(k) \rangle}, \tag{21}$$

By applying Equation (20) to the scene and target object in Figure 5, we obtain the following phrases:

$T = 1$	the
$T = 2$	the rectangle
$T = 3$	the green rectangle
$T = 4$	the large green rectangle
$T = 5$	the large light green rectangle
$T = 6$	the large vertical light green rectangle

These descriptions combine semantic and syntactic constraints. Word class sequences are chosen according to the bigram probabilities. Word choices are determined by best fit to the visual features of the target object.

The search process is implemented as an exhaustive search of all possible paths. For the longest utterances, with $K = 11$ word classes and $T = 6$ words, a total of $K^T = 1.77$ million utterances need to be evaluated. On a Pentium 1 GHz single processor machine, this takes approximately five seconds.

6.3. Contextual constraints

The simple utterances generated by the method described above can be ambiguous. Non-target objects in the scene might accidentally match the descriptions. To address this problem, we developed a measure of ambiguity of a target object description in the context of a set of competing objects. We start by defining the "fit" of an utterance to an object (ignoring context) as the product of the word-conditional pdfs evaluated for the features of the object, x:

$$\mathrm{fit}(x, Q) = \frac{\sum_{t=1}^{T} \log p(x|C(q_t))}{T}. \tag{22}$$

The denominator term normalizes the effect of length of Q. The ambiguity of a referring phrase is defined to be

$$\psi(Q) = \mathrm{fit}(x_{\mathrm{target}}, Q) - \max_{\forall x \neq x_{\mathrm{target}}} \mathrm{fit}(x, Q), \tag{23}$$

$\psi(Q)$ measures the fit of the utterance Q to the target object relative to the best competing object in the scene. The best competing object is defined as the object which is

TABLE VII. Utterances generated by combining syntactic and contextual constraints ($\alpha = 0.70$)

T	$\xi(Q)$	Utterance
1	-1.073×10^8	The
2	-1.159	The rectangle
3	2.786	The highest rectangle
4	2.042	The highest green rectangle
5	0.558	The highest vertical green rectangle
6	-0.235	The highest green vertical green rectangle

best described by Q. Syntactic and contextual constraints can be combined be defining a new score

$$\xi(Q) = \rho\gamma(Q) + (1-\rho)\psi(Q), \tag{24}$$

where ρ is an interpolation constant. As with Equation (19), we can find the utterance of length T which maximizes $\xi(Q)$ in order to generate descriptions of objects. We generated a new set of descriptive phrases again based on the scene and target object in Figure 5 with $\rho = 0.70$. This choice of ρ does not imply a bias against the ambiguity constraint, but instead compensates for differences of scale of $\gamma(\,)$ and $\psi(\,)$ inherent in the way each are computed.

Table VII shows the resulting utterances along with $\xi(Q)$ for each value of T.

A comparison between the descriptions with and without contextual constraints reveals the effect of including $\psi(Q)$ in the generation process. Based on bigram (syntactic) constraints, common color ("green") and size ("large") descriptors are preferred. Looking back at Figure 5, we see that other objects in the scene could also be described as large or green. Contextual considerations bias the system to select "highest" and "vertical" as less ambiguous terms. Bigram influences are still present in the context-sensitive phrases since the phrases are syntactically well-formed. Semantic constraints are also in effect since word classes are always mapped to words based on Equation (20) which assures semantic accuracy.

At this point we have reached our goal of generating object descriptions which integrate syntactic, semantic, and contextual constraints. The focus of this and the previous section has been on describing single objects (the equivalent of simple utterances in the training corpus). The next two sections describe the process of learning to generate complex utterances.

7. Learning relative spatial clauses

The original training corpus of 518 utterances contained 326 simple utterances (recall that we defined simple utterances to be utterances which refer to exactly one object). The remaining 37% of utterances were complex and referred to two or more objects. In this section, we describe methods for acquiring models which enable generation of utterances with relative spatial clauses ("the rectangle to the left of the red square"). The specific problems addressed are:

- Parsing complex utterances (from the training corpus) to identify subsequences of words which refer to objects.
- Establishing phrase-to-object correspondences. For example, in the utterance "the rectangle to the left of the red square," the learning system must decide which of the

two object descriptions ("the rectangle" or "the red square") refers to the target object, and which of the non-target objects serves as the referent of the remaining phrase.

- Acquiring visually grounded models of spatial words. New visual features will be introduced which measure relative angles and distances between pairs of objects.
- Learning a phrase-based bigram which models the syntax of complex utterances.

7.1. Parsing complex utterances using the acquired bigram language model

The bigrams described in Section 5.4 are estimated from truncated utterances. These bigrams are used as the basis for a probabilistic parser which identifies object phrases embedded in complex utterances.

We refer to the first set of bigrams as *object phrase bigrams*. A second set of *general bigrams* were trained using the entire untruncated training corpus. A set of bigrams may be thought of as a stochastic finite state automata (SFSA) in which each word is represented by a state and the bigram transition probabilities form arcs between states. Thus the two sets of bigram transition probabilities may be thought of as a pair of SFSAs. In order to construct a parser, we combine the SFSAs into one larger network. A simplified pair of networks is depicted in Figure 6. The general SFSA has three word states and utterance terminal nodes START$_g$ and END$_g$. The object phrase network consists of a subset of the words in the general network (since the vocabulary in the truncated corpus contains a subset of the full corpus). The dotted arcs indicate new transitions which are added to connect the two SFSAs. For each outgoing node in the general network which terminates in a word which is also part of the object phrase network, a new link is added with the same transition probability. Similarly, any state in the general network with an incoming arc from a word which is also part of the object phrase network receives a new transition of equal weight. Once all transitions are added, weights are rescaled to insure proper probabilities.

To use the SFSA as a phrase parser, the Viterbi algorithm (Rabiner, 1989) is used to find the most likely path through the network for a given utterance. A constant word

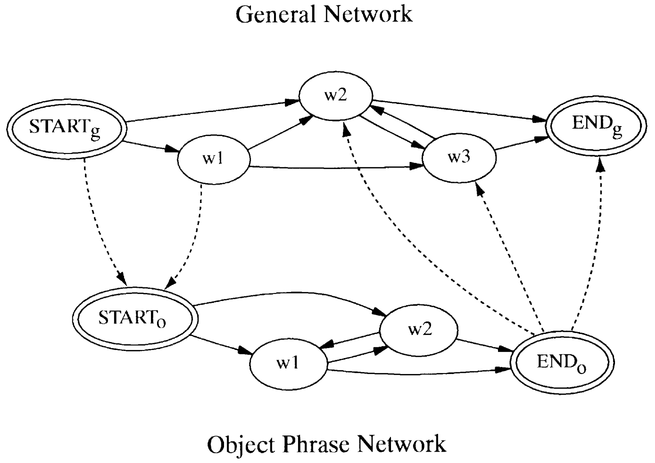


Figure 6. Stochastic finite state automata models of object phrases and complex utterances are combined to create a probabilistic phrase parser.

insertion penalty is inserted at each transition within the general network, but *not* within the object phrase network. Word insertion penalties are also inserted when transitioning into and out of the object phrase network (i.e. transitions to START_0 and from END_0). This configuration biases the Viterbi algorithm to align subsequences of the input to the object phrase SFSA whenever possible.

The output of the parser for some sample complex utterances is given below. Parentheses indicate groups of words which were aligned with the phrase SFSA:

1. **(the dark green rectangle)** above **(the light blue rectangle)**;
2. **(the purple rectangle)** to the left of **(the pink square)**;
3. the sea (**green rectangle**) to the right of **(the red rectangle)**;
4. **(the olive rectangle)** touching **(the green rectangle)**;
5. **(the tall green rectangle)** directly to the left of **(the large blue rectangle)**;
6. **(the purple rectangle)** between **(the red)** and blue rectangles;
7. **(the green rectangle)** to the right of the big **(pink rectangle)**.

The errors in examples 3 and 6 are due to new words in the complex utterances (“sea” and “rectangles”) which were not acquired in the phrase model. In example 7, “big” is also a new word but the parser is able to segment the remainder of the object phrase. The majority of complex utterances in the corpus were, however, parsed successfully. Although some errors were introduced at this stage, the probabilistic algorithms which operate on the output of the parser are nonetheless able to acquire useful structure.

In total, the parser found 184 complex utterances in the training corpus which contained exactly two object description phrases. The parsed utterances served as the basis for the next stage of processing.

7.2. Establishing phrase to object correspondence

Recall that the only input to the learning system is the utterance, the visual scene (i.e. the visual features extracted from each object in the scene), and the identity of the target object which was used to elicit the utterance. Each utterance selected by the parser, however, has two object phrases implying two distinct referent objects. Before relational terms may be acquired, the learning system must decide which of the two phrases refers to the original target object, and which of the remaining objects in the scene should be linked to the remaining phrase. We refer to the second referent as the *landmark object* or simply the landmark.

We will denote the two phrases which have been extracted from an utterance Q as Q_{p1} and Q_{p2}. The correspondence problem is solved using the following steps:

1. Select the target phrase. If $fit(x_{\mathrm{target}}, Q_{p1}) > fit(x_{\mathrm{target}}, Q_{p2})$ then decide that $Q_{\mathrm{target}} = Q_{p1}$ else $Q_{\mathrm{target}} = Q_{p2}$. The remaining unassigned phrase, by default, is assigned to be the landmark phrase Q_{landmark}.
2. Select the landmark object:

$$x_{\mathrm{landmark}} = \underset{\text{all objects } x \neq x_{\mathrm{target}} \text{ in scene}}{\arg\max} fit(x, Q_{\mathrm{landmark}}), \tag{25}$$

i.e. choose the object in the scene which is best described by the landmark phrase.

At this point, the target and landmark objects are identified, setting the basis for acquiring visually grounded models of spatial relation words and phrases.

7.3. Phrase tokenization

To illustrate the need for phrase tokenization, we can rewrite some sample training utterance with the target and landmark phrases tokenized (the previous step has determined which object phrase is which):

1. TARGET_PHRASE above LANDMARK_PHRASE
2. TARGET_PHRASE to the left of LANDMARK_PHRASE.
3. the sea TARGET_PHRASE to the right of LANDMARK_PHRASE.
4. TARGET_PHRASE touching LANDMARK_PHRASE.
5. TARGET_PHRASE directly to the left of LANDMARK_PHRASE.
6. TARGET_PHRASE to the right of the big LANDMARK_PHRASE.

Four of the utterances contain the spatial phrases "to the left/right of." A text filter was developed to detect 'stable' phrases and tokenize them. The token encodes the original word sequence which is required for generation. A simple iterative procedure is applied to the training utterances which looks for bigram transition probabilities above a preset threshold (we use 0.9 in all experiments). If $P(w_i|w_j)$ is greater than the threshold, all subsequences $(w_i|w_j)$ are replaced with the token $w_j - w_i$. This procedure is also applied to *reverse* bigrams, i.e. the probability of a word given the *next* word. The same threshold is applied to reverse bigrams. Once all stable word pairs have been tokenized, the tokenizer is rerun on the corpus iteratively until no further pairs are found. When run on the training corpus, four phrases were identified: "to-the-left-of," "to-the-right-of," "to-the," and "left-of."

7.4. Grounding spatial terms

Based on Regier's analysis (Regier, 1996), we chose three visual features to ground spatial semantics. Figure 7 illustrates these features. The *proximal distance* (prox_dist) is the distance between the landmark and target at the points where the objects are closest. In the figure, the dotted line segment connecting the corners of the objects is the proximal distance. The angle between this line and the horizon is the second feature,

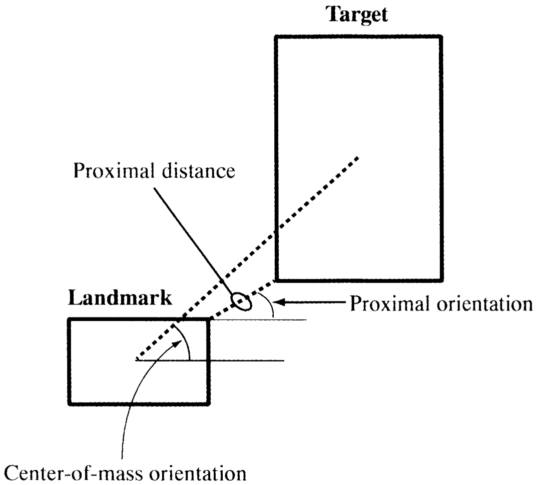


Figure 7. Visual features used to ground spatial semantics.

TABLE VIII. Word class and feature selection for spatial terms

Word class	Class members	Features
0	above, below, to-the-right-of, to-the-left-of, touching	prox_orient, com_orient, prox_dist
1	the	(ungrounded)
2	directly, and	(ungrounded)
3	horizontal vertical bright	(ungrounded)

proximal orientation (prox_orient). The third feature is the *center-of-mass orientation* (com_orient). If a line is drawn between the center of mass of the objects, the center-of-mass orientation is the angle this line makes relative to the horizon.

For each of the 184 training examples, the spatial features of the corresponding target–landmark pair were computed. The target and landmark phrases were removed from the training utterances (since they are already grounded). The remaining words in each utterance were paired with the three spatial terms and processed using the same learning procedures which we have described for learning object phrase acquisition. In other words, word class formation, feature selection, word-conditional density estimation (on the new features), and bigram models were constructed. The target and landmark phrases were used for phrase-based bigram estimation.

Table VIII lists the word classes which were acquired from the training corpus. All grounded terms are clustered into one class and use all three visual features. In other words, each of the terms listed in Cluster 0 have an associated three-dimensional Gaussian distribution which models the expected values of all three spatial features. All other words were automatically tagged as ungrounded.

Due to the small number of words involved, we decided to estimate word bigrams rather than word class bigrams. Figure 8 shows all word transitions with probabilities larger than 0.01.

At this point an important connection between word order and utterance semantics has been established in the model. Through the use of visual grounding, object phrases have been mapped to either target or landmark phrase tokens. The bigram language model shown in Figure 8 therefore differentiates phrases based on their *semantic role*, i.e. whether the phrase refers to a spatial landmark or the target. This connection between syntax and semantics will be necessary to generate complex utterances.

8. Generating relative spatial clauses

Before describing the spatial clause generation process, we address the question of how to decide when to generate a complex versus simple utterance. In the training corpus collected from the first speaker, 37% of the utterances are complex. One approach is to require the generation system to also generate complex utterances with the same frequency. We would like the system to generate complex rather than simple utterances when the system's confidence in the best possible simple utterance is low. We accomplished this by using the object phrase generation system to produce simple utterances for a set of 200 novel scenes. For each utterance Q generated by the system, we evaluated the context-sensitive score of the utterance $\xi(Q)$ (Equation (24)). The scores were accumulated in a histogram. We then found the 37% threshold, i.e. the score threshold below which the 37% of simple utterance scores lie. This threshold is denoted τ. Based on the training corpus, we found $\tau = 0.71$. Given a novel scene, the system first

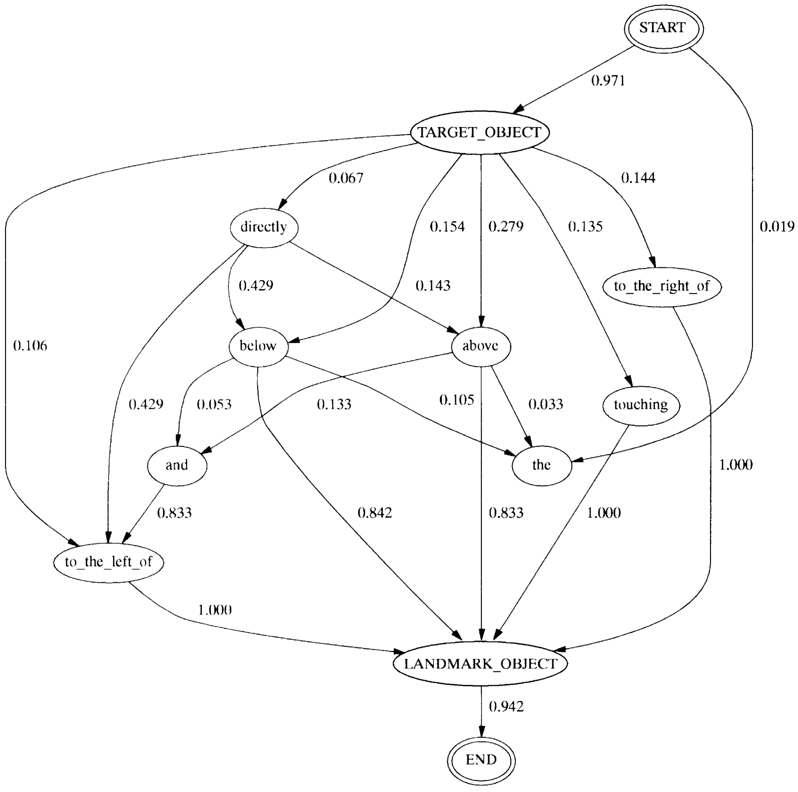


Figure 8. Word-class and phrase based statistical bigram for complex utterances with relative spatial clauses.

generates a simple utterance Q. If $\xi(Q) > \tau$ then the system outputs Q as its final output. If $\xi(Q) \leqslant \tau$ then a complex utterance is generated using the method presented below.

The goal of generating relative spatial clauses ("touching the pink rectangle") is to reduce ambiguity of the target object. Given the target object, the system must select a landmark which is easy to describe using the grounded object language models that have been learned, and lies in a location which is easy to describe using the acquired spatial language. The selection of a landmark is achieved by first generating simple utterances for each object in the scene (other than the target). The context-sensitive score $\xi(\,)$ is computed for each utterance. The threshold τ (defined above) is used to select which of the potential landmarks can be unambiguously described using a simple utterance. If none of the objects are describable with sufficient confidence (i.e. $\xi(Q) < \tau$ for all objects), then the relative clause generator fails and the system is forced to generate a simple utterance. Assuming one or more potential landmarks can be described, the three spatial features are extracted for each candidate relative to the target. For each potential landmark, the best fitting spatial term Q_{spatial} is selected. The candidate landmark for which $\gamma(Q)$ is highest is selected as the landmark.

At this point the system has three sequences of words: (1) a simple utterance describing the landmark, (2) a spatial term which describes the target–landmark relation, and (3) a simple utterance describing the target. Given these components, a dynamic programming algorithm uses the phrase level bigrams (Figure 8) to find the most likely order in which to sequence the words to form a complex utterance.

Figure 9 shows representative output from the final system for several randomly generated scenes which were not part of the training corpus. The target object specified to the system is indicated in each images with an arrow. For seven of the targets the system decided to generate simple utterances. In five cases it was unable to generate an unambiguous simple utterances and instead opted for a complex utterance.

9. Text-to-speech conversion

The text output from the generation system is synthesized using whole word concatenation. The corpus of speech recordings from the training corpus is used as the basis for concatenative speech synthesis. Since only the words used in the training corpus can be acquired by the system, we are able to use whole words as synthesis units.

To support concatenative synthesis, the speech corpus was automatically aligned to the text transcripts using the Viterbi algorithm using context-dependent triphone acoustic models and a phonetic dictionary with full coverage of the training corpus vocabulary (Yoder, 2001). The output the alignment process is a set of word indices which specify start and end samples of each word in the source speech recordings.

Synthesis of a novel utterance consists of finding the set of speech segments in the corpus which minimizes the number of jumps from one point in a speech recording to another. The Viterbi algorithm is used to efficiently solve this search problem.

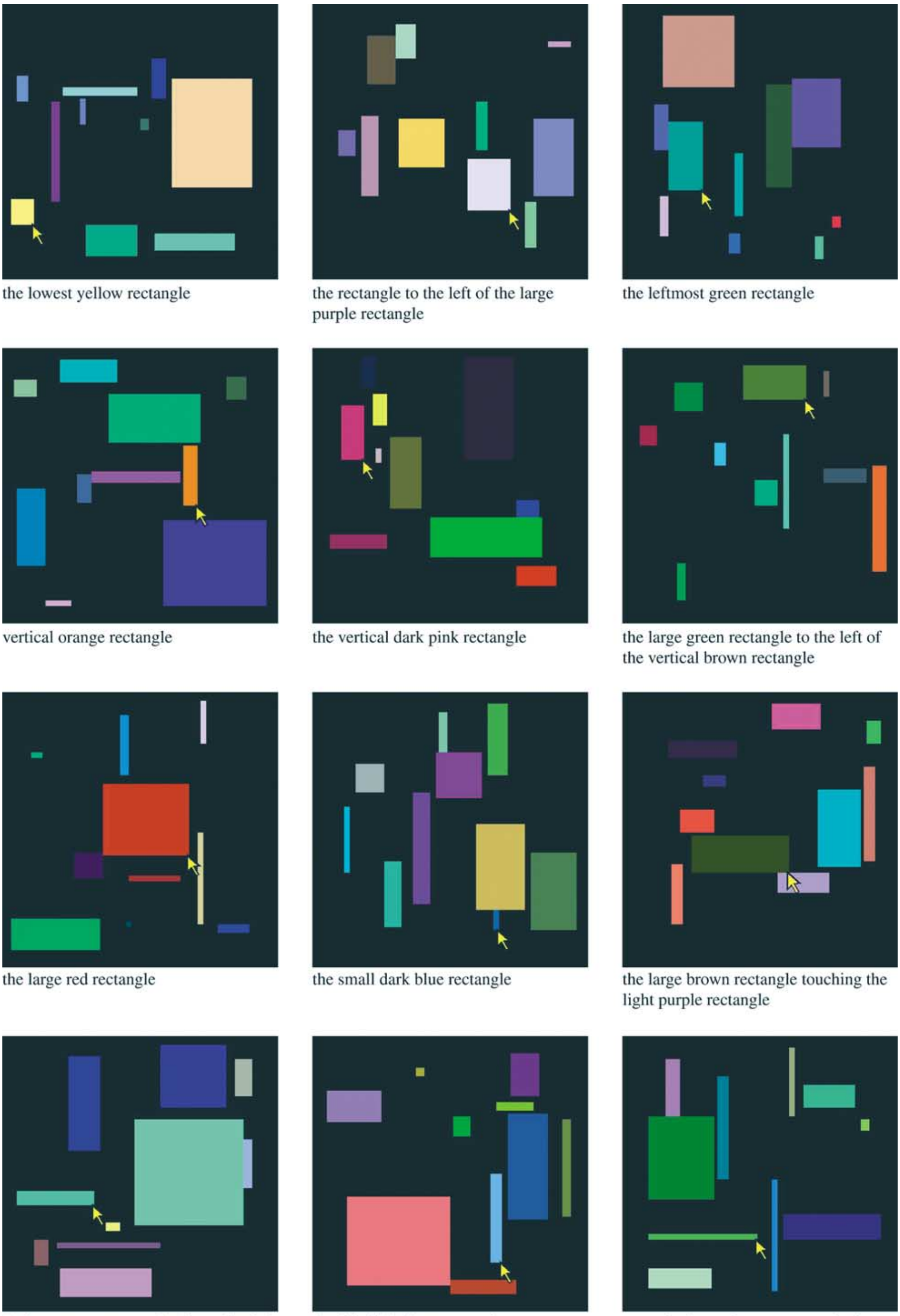
Although no attempt is made to smooth the points of concatenation in the final output, subjects in evaluations reported that the spoken utterances were usually highly intelligible although often lacking natural prosody. Clearly, many existing techniques may be applied to improve the quality of synthesis but for the task at hand our simple approach was sufficient.

10. Evaluation

We evaluated spoken descriptions from the original human-generated training corpus and from the output of the generation system. Three human judges unfamiliar with the technical details of the generation system participated in the evaluation. An evaluation program was written which presents images on a computer screen paired with spoken descriptions which are heard through speakers. The evaluation was a forced choice task. Judges were asked to select the rectangle which best fit the description by clicking on the object using a mouse pointer. A ‘play-again’ option was provided in the interface that allowed judges to listen to spoken descriptions multiple times if desired before making a selection.

The goal of the evaluation was to measure the level of semantic accuracy and ambiguity of the descriptions. We did not explicitly evaluate the naturalness of the synthetic speech. Implicitly, however, the intelligibility of the synthesis was evaluated since low intelligibility should result in low understandability.

Each judge evaluated 200 human-generated and 200 machine-generated spoken descriptions. All judges evaluated the same sets of utterances. Responses were evaluated by comparing the selected object for each image to the actual target object which was selected in order to produce the verbal description. Table IX shows the results for both human-generated and machine-generated results.



the lowest yellow rectangle

the rectangle to the left of the large purple rectangle

the leftmost green rectangle

vertical orange rectangle

the vertical dark pink rectangle

the large green rectangle to the left of the vertical brown rectangle

the large red rectangle

the small dark blue rectangle

the large brown rectangle touching the light purple rectangle

the large green rectangle above the dark brown rectangle

the thin light blue rectangle

green horizontal green rectangle above the lowest green rectangle

Figure 9. Sample output from the generation system.

Table IX. Results of an evaluation of human- and machine-generated descriptions

Judge	Human-generated (% correct)	Machine-generated (% correct)
A	90.0	81.5
B	91.2	83.0
C	88.2	79.5
Average	89.8	81.3

Averaged across the three listeners, the original human-generated descriptions were correctly understood 89.8% of the time. This result reflects the inherent difficultly of the rectangle task. An analysis of the errors reveals that a difference in intended versus inferred referents sometimes hinged on subtle differences in the speaker and listener's conception of a term. For example the use of the terms "pink," "dark pink," "purple," "light purple," and "red" often lead to comprehension errors. In some cases it appears that the speaker did not consider a second object in the scene which matched the description he produced.

The average listener performance on the machine-generated descriptions was 81.3%, i.e. a difference of only 8.5% compared to the results with the human-generated set. An analysis of errors reveals that the same causes of errors found with the human set also were at play with the machine data. Differences in intended versus inferred meaning hinged on single descriptive terms. In some cases, an object was labeled using a descriptive term which was chosen mainly for its effect in reducing ambiguity rather than for its description accuracy. This lead at times to confusions for listeners. In addition, we also found that the system acquired an incorrect grounded model of the spatial term "to-the-left-of" which lead to some generation errors. This would easily be resolved by providing additional training examples which exemplify proper use of the phrase.

The results presented in this section demonstrate the effectiveness of the learning algorithms to acquire and apply grounded structures for the visual description task. The semantics of individual words and the stochastic generation methods were able to produce natural spoken utterances which human listeners were able to understand with accuracies only 8.5% lower than original utterances spoken from the training corpus.

11. Discussion

Language gains its power from its generative capacity. A finite vocabulary of words may be combined to form vast numbers of unique word sequences. From a language learning perspective, a key challenge is to develop algorithms which can generalize from training examples so that novel word sequences may be generated as needed. DESCRIBER achieves this goal. It is able to describe scenes it has not encountered during training, and will often choose sequences of words which never occurred in the training data. This generative capacity is a result of the formation and use of word classes and phrase structure. Consider the role of acquired word classes. Statistical rules of word order acquired from observation of some words are mapped to other words on the basis of shared class membership. For example, if the sequence 'large blue square' is observed, the sequence 'small red rectangle' can be generated if the appropriate word classes have been acquired. Since word classes are formed partially on the basis of semantic similarity, bottom-up visual grounding directly influences the application of syntactic rules to words. Thus, the rules of symbol manipulation in DESCRIBER are influenced by subsymbolic, visually grounded structure.

The scope of the current results is limited in several ways with respect to our long-term goal of creating domain-independent trainable systems. These limitations highlight challenging problems which must be addressed in the future if the goal is to be achieved.

The visual scenes processed in DESCRIBER are synthetic and highly constrained. Objects are all rectangular, of constant color, and guaranteed to be non-overlapping. Since the scenes are computer-generated, visual features derived from the scenes are noise-free. For operation with complex real world input such as that derived from a computer vision system, robustness to various sources of perceptual ambiguity and noise must be addressed.

The syntactic structures acquired by DESCRIBER are not recursive. The layered Markov model structures used in DESCRIBER can be extended to higher levels of embedding, but cannot represent arbitrary levels of recursion. To do so, statistical context free grammars or functional equivalents would need to be introduced. Acquisition of recursive structures would require exploration of different learning strategies.

The training process was simplified by using utterance truncation to focus initial learning on simple object descriptions (i.e. without optional relative spatial phrases). More complex training utterances including those with relative spatial clauses were introduced in a second phase. Without this two-stage procedure, the system would have failed to learn from the available number of training examples. To avoid this domain-specific simplification, a model of attention is required which focuses learning on simple examples before considering complex input. This, in turn, requires automatic domain-independent classification of simple versus complex training examples.

The output of DESCRIBER's learning algorithms depends on a small number of parameters which were set manually. These include α and T used for hybrid word class creation, the bigram transition probability threshold used for tokenizing phrases, and ρ used to balance syntactic and contextual constraints during generation. These parameters have been manually adjusted for optimal system performance. A desirable but difficult extension of this work would be to automate the optimization of these parameters. The optimization process would require performance-based reinforcement feedback. In essence, the system would attempt to put its linguistic knowledge to use, and use environmental feedback to adjust these parameters.

12. Conclusions and future directions

Our goal in developing DESCRIBER was to explore the use of learning algorithms to infer scene-to-language mappings from show-and-tell input. In this paper, we have presented the underlying algorithms, implementation, and evaluation of a system which generates natural spoken descriptions of objects in visual scenes. Learning algorithms process training examples (images paired with natural language descriptions) to acquire structures which link linguistic and visual representations.

Visually grounded language learning occurs in several stages. In the first part of the paper, we described the acquisition and use of structures for referring to single objects. We showed that distributional analysis of word co-occurrence patterns can be combined with semantic associations to form word classes. A feature selection algorithm assigns visual features to word classes based on an information-theoretic analysis. The semantics of individual words are grounded in terms of selected features by estimating

Gaussian word-conditional probability density functions. A word-class bigram language model is acquired as a model of word order. These components are used in a generation algorithm which integrates syntactic, semantic, and contextual constraints to produce optimal natural language descriptions of objects.

In the next sections of the paper, we described methods for parsing complex training utterances using acquired class-based language models. The parsed utterances are automatically brought into semantic correspondence with objects from the training images. The correspondence process leverages grounding models acquired in the first stage of learning. Grounded semantics are then acquired for spatial terms, and a phrase-level bigram language model of complex utterances is learned. Using these new structures, a modified generation system is able to generate complex utterances which include an expression referring to the target object, as well as a relative spatial clause using an automatically selected landmark. A context-sensitive score of utterance ambiguity drives the decision of when to generate simple versus complex verbal descriptions.

The system was evaluated by three listeners. Each listener was asked to select the most likely target within a visual scene given a verbal description. The same procedure was repeated with these listeners on a subset of the original human-generated training utterances. Human produced utterances led to correct object selections 89.8% of the time whereas machine-generated utterances led to correct selections 81.3% of the time. Thus the system is able to communicate its 'intent' via natural language with near human level precision. This result demonstrates the viability of learning-based approaches for grounded language generation.

There are several directions we plan to pursue to extend this work. The task chosen for this initial investigation is of a highly abstract nature, but the underlying algorithms can be applied to numerous practical applications. In the future, we will replace synthetic visual features with features derived from video images, and we will also experiment with tasks involving more complex linguistic constructions. We will also experiment with 'inverting' the acquired linguistic structures to enable visually grounded speech understanding.

We began this paper by motivating our work as a method of developing language generation systems for applications such as sports commentators and navigation systems. The work presented here represents our first steps in this direction. Although many significant challenges remain, we believe that a teach by-example approach is a feasible, and ultimately more flexible, approach to visually grounded language generation.

The author is grateful to Janet Cahn, Peter Gorniak, Kai-Yuh Hsiao, Yuri Ivanov, Tony Jebara, Rupal Patel, Owen Rambow, and anonymous reviewers for helpful comments which helped shape the presentation of this paper.

References

André, E. & Rist, T. (1995). Generating coherent presentations employing textual and visual material. *Artificial Intelligence Review* **9**.

Barsalou, L. (1999). Perceptual symbol systems. *Behavioural and Brain Sciences* **22**, 577–609.

Braine, M. D. (1971). On two types of models of the internalization of grammars. In *The Ontogenesis of Grammar* (D. Slobin, Ed),. Academic Press, New York.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*, Wiley–Interscience, New York, NY.

Dale, R. (1992). *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*, MIT Press, Cambridge, MA.

Dale, R. & Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science* **19(2)**, 233–263.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika* **40**, 237–264.

Gorin, A. L. (1995). On automated language acquisition. *Journal of the Acoustic Society of America* **97(6)**, 3441–3461.

Harnad, S. (1990). The symbol grounding problem. *Physica D* **42**, 335–346.

Herzog, G. & Wazinski, P. (1994). VIsual TRAnslator: linking perceptions and natural language descriptions. *Artificial Intelligence Review* **8**, 175–187.

Jordan, P. & Walker, M. (2000). Learning attribute selections for nonpronominal expressions. *Proceedings of ACL.*

Lakoff, G. & Johnson, M. (1980). *Metaphors We Live By*, University of Chicago Press, Chicago.

Markman, E. M. (1991). *Categorization and Naming in Children*, MIT Press, Cambridge, MA.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77(2)**, 257–285.

Regier, T. (1996). *The Human Semantic Potential*, MIT Press, Cambridge, MA.

Roy, D. (2000). Grounded speech communication. *Proceedings of the International Conference on Spoken Language Processing.*

Roy, D. (2000/2001). Learning visually grounded words and syntax of natural spoken language. *Evolution of Communication* **4(1)**.

Roy, D. (in press). Grounded spoken language acquisition: experiments in word learning. *IEEE Transactions on Multimedia.*

Roy, D. & Pentland, A. (2002). Learning words from sights and sounds: a computational model. *Cognitive Science* **26(1)**, 113–146.

Siskind, J. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Artificial Intelligence Review* **15**, 31–90.

Therrien, C. (1989). *Decision, Estimation and Classification*, Wiley, New York.

Yoder, B. (2001). *Spontaneous speech recognition using hidden markov models.* Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.

(Received 10 October 2001 and accepted for publication 8 April 2002)

Appendix A. Vocabulary in training corpus

Vocabulary and token counts in the complete (untruncated) transcriptions of the rectangle task corpus (83 unique token types).

920 the
774 rectangle
174 green
139 blue
125 purple
118 large
85 pink
80 to
76 of
63 square
63 brown
61 light
59 horizontal
53 small
53 vertical
46 dark
45 above
44 left

19 grey
18 rectangles
16 white
16 tall
15 olive
13 thin
13 directly
12 rightmost
12 and
11 leftmost
11 highest
11 salmon
10 off
10 colored
7 between
7 maroon
6 two
5 other

3 florescent
3 closest
3 longest
2 narrow
2 long
2 biggest
2 gold
2 smaller
2 thinnest
2 near
2 skinniest
2 skinny
1 peach
1 sky
1 tan
1 lower
1 furthest
1 another

40	yellow
39	orange
37	red
36	below
35	right
33	touching
32	bright
22	smallest
22	lowest
21	largest
4	faded
4	sea
4	big
4	brightest
4	teal
4	black
4	larger
4	uppermost
3	tiny
3	flat
1	tallest
1	violet
1	surrounded
1	by
1	cream
1	very
1	three
1	bark
1	shorter

Vocabulary and token counts in the truncated transcriptions of the rectangle task corpus (70 unique token types).

664	the
620	rectangle
133	green
112	blue
99	purple
96	large
59	pink
58	horizontal
52	light
50	vertical
49	small
48	square
46	brown
45	dark
36	yellow
27	orange
26	red
25	bright
22	lowest
20	smallest
18	largest
17	grey
15	tall
13	thin
13	olive
12	rightmost
11	white
11	leftmost
11	highest
10	colored
9	salmon
5	off
4	faded
4	sea
4	brightest
4	teal
4	black
4	larger
4	maroon
4	uppermost
3	flat
3	florescent
3	longest
3	rectangles
2	narrow
2	tiny
2	big
2	long
2	biggest
2	gold
2	to
2	of
2	smaller
2	thinnest
2	skinniest
2	left
2	other
1	above
1	right
1	tan
1	lower
1	tallest
1	violet
1	between
1	cream
1	very
1	two
1	skinny
1	three
1	shorter