Combining symbolic and corpus-based approaches for the generation of successful referring expressions

Konstantina Garoufi and **Alexander Koller**
Area of Excellence "Cognitive Sciences"
University of Potsdam, Germany
{garoufi, akoller}@uni-potsdam.de

Abstract

We present an approach to the generation of referring expressions (REs) which computes the unique RE that it predicts to be fastest for the hearer to resolve. The system operates by learning a maximum entropy model for referential success from a corpus and using the model's weights as costs in a metric planning problem. Our system outperforms the baselines both on predicted RE success and on similarity to human-produced successful REs. A task-based evaluation in the context of the GIVE-2.5 Challenge on Generating Instructions in Virtual Environments verifies the higher RE success scores of the system.

1 Introduction

The generation of referring expressions (REs) is one of the best-studied problems in natural language generation (NLG). Traditional approaches (Dale and Reiter, 1995) have focused on defining the range of possible valid REs (e.g., as those REs that describe the target object uniquely) and on simple heuristics for choosing one valid RE (e.g., minimal REs). Recently, the question of how to choose the best RE out of the possible ones has gained increasing attention (Krahmer et al., 2003; Viethen et al., 2008). This process has been accelerated by the systematic evaluation of RE generation systems in the context of RE generation challenges (Belz and Gatt, 2007; Gatt and Belz, 2010).

Almost all of these approaches optimize the *humanlikeness* of the NLG system, i.e. the similarity between system-generated REs and human-generated REs from some corpus. However, in order to be most helpful to the user, an NLG system should arguably produce REs that are *easy to understand*. As Belz and Gatt (2008) show, these are not the same: In particular, the scores for humanlikeness and usefulness in task-based evaluations of systems participating in the TUNA RE generation challenge are not correlated. It would therefore be desirable to optimize a system directly for usefulness.

A second characteristic of most existing RE generation systems is that they are limited to generating single noun phrases in isolation. By contrast, planning-based approaches (Appelt, 1985; Stone et al., 2003; Koller and Stone, 2007) generate REs in the context of an entire sentence or even discourse (Garoufi and Koller, 2010), and can therefore exploit and manipulate the linguistic and non-linguistic context in order to produce succinct REs (Stone and Webber, 1998). However, these approaches have not been combined with corpus-based measures of humanlikeness or understandability of REs.

In this paper, we present the mSCRISP system, which extends the planning-based approach to NLG with a statistical model of RE understandability. mSCRISP uses a metric planner (Hoffmann, 2002) to compute the best REs that refer uniquely to the target referent, and thus combines statistical and symbolic reasoning. We obtain the cost model by training a maximum entropy (maxent) classifier on a corpus of human-generated instruction giving sessions (Gargett et al., 2010) in which every RE can be automatically annotated with a measure of how easy it was for the hearer to resolve. Although mSCRISP is in principle capable of generating complete in-

Proceedings of the 13th European Workshop on Natural Language Generation (ENLG), pages 121–131,
Nancy, France, September 2011.

struction discourses, we only evaluate its RE generation component here. It turns out that mSCRISP generates more understandable REs than a purely symbolic baseline, according to our model's estimation of understandability. Furthermore, mSCRISP generates REs that are more similar to high-quality human-generated REs than either the symbolic or a purely statistical baseline. Finally, a full task-based evaluation in the context of the GIVE-2.5 Challenge on Generating Instructions in Virtual Environments[1] (Koller et al., 2010; Striegnitz et al., 2011) verifies the higher referential success of the system.

Plan of the paper. We first compare our model to earlier work in Section 2. We then introduce the planning-based approach to NLG on which mSCRISP is based in Section 3. Section 4 lays out how we obtain a maximum entropy model of RE attribute preferences from our corpus, and Section 5 shows how we bring the two approaches together using metric planning. We present the evaluation in Section 6 and conclude in Section 7.

2 Related work

Our work stands in a recent tradition of approaches that attempt to learn optimal RE generation strategies from corpora. For instance, Viethen et al. (2008) tune the parameters of the graph-based algorithm of Krahmer et al. (2003) by learning attribute costs from the TUNA corpus (Gatt et al., 2007). Stoia et al. (2006) share with us a focus on situated generation in a virtual environment. They train a decision tree learner using a wide range of context features, including dialog history, spatio-visual information and features capturing relations between objects in the scene. The context features we use in this paper are partially inspired by theirs. However, our work differs from this line of research in that we do not primarily attempt to replicate the REs produced by humans, but to train a system to produce REs that are easy to understand by humans.

There are a number of related systems which optimize for understandability. Paraboni et al. (2007) present two rule-based RE generation systems which can deliberately produce redundant REs, and evaluate these systems to show that they outperform earlier systems in terms of understandability. On the other hand, their approach is not corpus-based and is therefore harder to fine-tune to the communicative needs of hearers using empirically determined parameters. Golland et al. (2010) present a maximum entropy model which acts optimally with respect to a hearer model; but their system is focussed on spatial descriptions of objects in non-dynamic scenes. Furthermore, dialogue and NLG systems based on reinforcement learning optimize their expected utility for human or simulated users. However, because of the complexity of reinforcement learning, this has for the greatest part been applied to RE generation only in the most rudimentary way, e.g. to distinguish whether or not to use jargon in a technical dialogue (Janarthanam and Lemon, 2010). Decision-making problems of a broader scope have started getting addressed by such techniques only very recently (Dethlefs et al., 2011).

Finally, NLG systems based on planning, such as Koller and Stone (2007), typically optimize for RE size instead of either humanlikeness or understandability. One exception is Bauer and Koller (2010), where sentence generation with a probabilistic grammar formalism is performed using a metric planner. That work generates REs which are probable and therefore in a certain sense humanlike; yet it focuses on syntactic choice and does not take understandability into account, neither has it been evaluated on RE generation tasks.

3 Planning utterances in situated context

We build upon CRISP (Koller and Stone, 2007), a planning-based NLG model which encodes sentence generation with tree-adjoining grammars (TAG; (Joshi and Schabes, 1997)) as an automated planning problem. The CRISP model solves the problem of translating a given communicative goal into a complete natural language sentence in a single step. Although we only use CRISP to generate REs that are individual noun phrases here, these are in fact part of a comprehensive integrated sentence planning and realization process, which has also been extended to the generation of entire discourses of navigation instructions (Garoufi and Koller, 2010).

CRISP assumes a TAG lexicon in which each elementary tree has been enriched with semantic and

[1] http://www.give-challenge.org/research/page.php?id=give-2.5-index

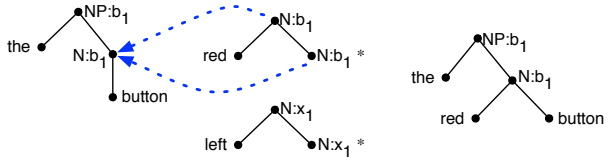

Figure 1: A simplified example of a CRISP lexicon and the derivation of the RE "the red button" describing b_1.

pragmatic information in addition to the syntactic information it encodes. The generator obtains awareness of the domain entities a hearer knows about, their semantic content and the relations holding between them by tapping into a knowledge base that models the scene. It then generates REs for these entities by reasoning about how its lexicon entries can be combined into well-formed derivation trees that amount to correct and distinguishing descriptions of the referents. Given an example knowledge base $\{\mathsf{button}(b_1),\ \mathsf{red}(b_1),\ \mathsf{button}(b_2),\ \mathsf{blue}(b_2),\ \mathsf{left\text{–}of}(b_2, b_1))\}$, and a communicative goal that involves describing b_1, Figure 1 shows with a simplified version of CRISP's lexicon how the derivation of "the red button" referring to b_1 is performed.

In order to generate this RE, CRISP converts the lexicon of Figure 1 and the given communicative goal into a planning problem, whose operators are shown in simplified form in Figure 2. Preconditions of an operator determine which logical propositions must be true in a given state so that the operator can be executed, while its effects specify how the truth conditions of these propositions change after the execution. It is important to notice that both syntactic preconditions and effects (e.g., subst specifies open substitution nodes, ref connects syntax nodes to the semantic individuals to which they refer, and canadjoin indicates the possibility of an auxiliary tree adjoining the node) and semantic ones are integrated into these operators. In particular, **red** includes a precondition $\mathsf{red}(x)$, whereas **left** includes a more complex precondition estimating the eligibility of an entity to be described as "left" at a given state of the derivation. This way CRISP ensures that the attributes selected are applicable to the entities described and that the resulting REs are correct.

The planning problem adopts the facts of the knowledge base in its initial state and sets as its goal the fulfillment of the communicative goal along

red(u, x):
Precond: $\mathsf{canadjoin}(\mathrm{N}, u), \mathsf{ref}(u, x), \mathsf{red}(x), \ldots$
Effect: $\forall y.\neg\mathsf{red}(y) \rightarrow \neg\mathsf{distractor}(u, y), \ldots$

left(u, x):
Precond: $\forall y.\neg(\mathsf{distractor}(u, y) \wedge \mathsf{left\text{–}of}(y, x)),$
$\mathsf{canadjoin}(\mathsf{N}, u), \mathsf{ref}(u, x), \ldots$
Effect: $\forall y.(\mathsf{left\text{–}of}(x, y) \rightarrow \neg\mathsf{distractor}(u, y)), \ldots$

the–button(u, x):
Precond: $\mathsf{subst}(\mathsf{NP}, u), \mathsf{ref}(u, x), \mathsf{button}(x), \ldots$
Effect: $\forall y.(\neg\mathsf{button}(y) \rightarrow \neg\mathsf{distractor}(u, y)),$
$\neg\mathsf{subst}(\mathsf{NP}, u), \ldots$

Figure 2: Simplified CRISP planning operators for the lexicon of Figure 1.

with the satisfaction of a set of syntactic and semantic constraints. The former encode syntactic completeness of the derivation while the latter are specified as $\forall u \forall x.\neg\mathsf{distractor}(u, x)$, conveying that a complete derivation tree must eliminate all possible distractors from any entities it refers to, thus making sure that all generated REs are distinguishing. With these constraints, it is easy to examine what reasoning CRISP follows for the generation of an RE describing b_1. Having executed the action **the–button**(n_1, b_1), it can eliminate all entities of the domain that are not buttons from the set of distractors for b_1. However, the button b_2 in the domain remains as a distractor. To change this, CRISP goes on to check the preconditions of other available operators. It finds that even though **left**(n_1, b_1) is not applicable, as b_2 and not b_1 is the leftmost button in the scene, **red**(n_1, b_1) is. Since this operator now eliminates b_2 (which is blue) as a distractor, the goals have been achieved and the planner terminates.

4 A maxent model for successfulness

We now present how to obtain a corpus which allows to determine how fast a hearer understood an RE, and discuss how to train a maxent model that predicts this.

4.1 RE attributes in the GIVE-2 corpus

We use the GIVE-2 corpus of Giving Instructions in Virtual Environments[2] (Gargett et al., 2010), which

[2] http://www.give-challenge.org/research/page.php?id=give-2-corpus

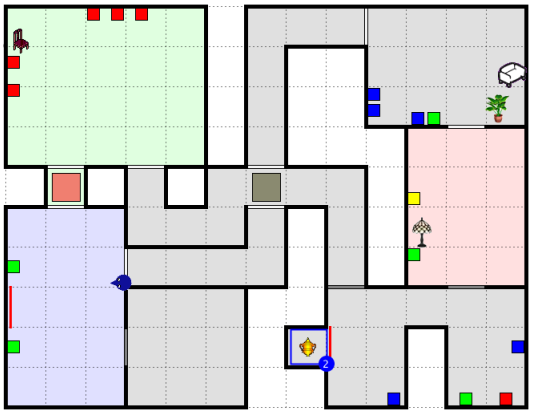

Figure 3: Map of a virtual world from the GIVE-2 corpus.

RE attribute type	%
Absolute property (color; e.g. "red")	79.83
Taxonomic property (type; e.g. "button")	59.80
Viewer-centered (e.g. "on the right", "the left one")	19.33
Micro-level landmark intrinsic (e.g. "by the chair")	17.37
Macro-level landmark intrinsic (e.g. "next to the doorway")	8.54
Distractor intrinsic (e.g. "next to the yellow button")	7.00

Table 1: The six most frequent attribute types in the English edition of the GIVE-2 corpus.

consists of instruction giving sessions in 3D virtual worlds. In these sessions a human instruction giver (IG) guides a human instruction follower (IF) through the world with the goal of completing a treasure hunting task. Although the worlds feature varied types of objects (e.g. movable objects such as chairs and immovable features of rooms such as doorways), instruction followers can directly manipulate only one type of targets before picking up the treasure, which is buttons. Figure 3 presents a bottom-up view of one of the three corpus worlds.

Gargett et al. have annotated the expressions referring to button targets of manipulation in the corpus with the types of attributes of which they are made up. In this work we focus on the six most frequent attribute types, shown in Table 1. Notice that each attribute type is a semantic concept which may be realized in different ways, according to the properties of the referent. We refer to the resulting realizations as attributes (e.g. "red" and "blue" are attributes of the type "absolute"). Of the 714 annotated REs in the English edition of the GIVE-2 corpus, 598 only use attributes of the above six types.

4.2 Successfulness of REs

Annotated REs in the GIVE-2 corpus are issued by the human IGs in order to help their partners identify targets of manipulation in the world. In this task-based setting, we can assess whether an RE has served its purpose with success or not by determining whether it leads the IF to manipulating the intended referent. A manual annotation of RE success reveals that 92% of all human-produced REs in the corpus allow the IF to correctly identify the referent.

This task-based success measure could be a good candidate for determining the understandability of a RE, except that data in which one class accounts for 92% of all instances is too skewed to be useful for machine learning. We can achieve a more even split of the data by assuming that an IF who understands the RE easily will walk towards the correct referent quickly and directly; in other words, the average speed at which they approach the referent is a measure of understandability. We define the *successfulness* $succ(r)$ of an RE r as follows:

$$succ(r) = \begin{cases} 0 & \text{if } r \text{ was not correctly resolved} \\ \frac{\Delta S}{\Delta T} & \text{otherwise,} \end{cases}$$

where ΔS is the distance in the GIVE world (including turning distance) between the target referent and the hearer's location at the time when they are presented with the RE, and ΔT is the time elapsed between the presentation of the RE and the manipulation of the referent. We can now split the REs in the corpus into a class of high successfulness and one of low successfulness as follows:

$$succ^*(r) = \begin{cases} 0 & \text{if } succ(r) \leq \tilde{S} \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

where $\tilde{S}$ is the median of all values that $succ(r)$ takes for all REs r in the data. This *binarized successfulness* abstracts away from the exact numeric value of an RE's successfulness, which is not important for our purpose, and allows us to create a balanced dataset with two classes of equal size.

4.3 Context features

We assume that in any given context, all attributes of the same type are equally easy to understand for a hearer. However, we do not assume that the same attribute types are easy to understand (i.e., have high successfulness) in all possible contexts. A color attribute may be easier to understand in a scene where there are no distractors of the same color as the referent—not just because it is conspicuous, but also because the hearer will not be visually distracted by similar distractors. Conversely, if a visually salient landmark is available for describing the target referent, it might be harder to process the referent's color than its location relative to the landmark (Viethen and Dale, 2008).

We model this connection of the RE resolution process with the currently visible scene through a collection of ten *context features*, which we list in Table 2. For our experiments, we extract most of these features from the corpus automatically, except for the *Round* and *ReferenceAttempt* features, which we annotated manually. For each object relations and referent's distinctiveness feature, we consider as scope of comparison (near the referent, in the referent's room or in the whole virtual world) the one that yields best results in Subsection 6.1. Note that some context features (such as *MicroLandmarkInRoom*) take binary values, whereas others (e.g. *Angle*) take a range of numeric values.

4.4 The maximum entropy model

Now we combine the information we have about human RE choices, the context in which they were issued and their relative successfulness in order to train a maximum entropy model that can estimate the successfulness of any RE in any context. We model an RE r as a set of attributes and let $a_j(r) = 1$ (where $j = 1, \ldots, 6$) iff r contains an attribute of type a_j. We further assume that $c_i(s)$ (where $i = 1, \ldots, 10$) takes the value of the feature c_i on the scene s, and combine attributes and context features into derived features of the form

$$\phi_{ij}(r, s) = c_i(s) \cdot a_j(r).$$

The derived features allow us to cast the problem as a simple binary classification task, in which our goal is to estimate the conditional probability of an RE r issued in a scene s being successful, given a joint representation of attributes and context:

$$P\left(succ^*\left(r\right) = 1 \mid \{\phi_{ij}\left(r, s\right)\}_{i,j}\right)$$

We train a maximum entropy model to learn this distribution. This choice of model has several advantages; among others, that we can later convert the model parameters into parameters for a planning model quite easily (see Section 5). For training we use the logistic regression implementation of the Weka data mining workbench (Hall et al., 2009). The model estimates the above probability as:

$$\hat{P}(succ^*(r) = 0 \mid \{\phi_{ij}(r, s)\}_{i,j}) = \frac{1}{1 + e^{-z(r,s)}}, \quad (2)$$

where $z(r, s) = \sum_{i,j}(w_{ij} \cdot \phi_{ij}(r, s)) + w_0$ for model coefficients w_{ij} and intercept w_0. By letting $v_j(s) = \sum_i(w_{ij} \cdot c_i(s))$, we can rewrite this equation as $z(r, s) = \sum_j(v_j(s) \cdot a_j(r)) + w_0$. In this way, we can obtain *attribute weights* $v_j(s)$ for each attribute type a_j. Notice that the weight of an attribute type depends on the current scene s (as seen through the context features). In our data, we observe that every context feature in Table 2 affects the weight of at least one attribute type.

5 Optimizing successfulness using metric planning

We can now describe the mSCRISP system, which combines the planning-based NLG algorithm from Section 3 with the maxent model for assigning successfulness estimates to REs from Section 4. We employ for this the formalism of *metric planning* (Fox and Long, 2003), which we use to assign to each planning operator a *cost*. The cost of a plan is the sum of the costs of the actions that were used in it, and a metric planner will try to find a plan of minimal total cost. Because the original planning problem already enforces that an RE must refer uniquely, this amounts to finding the RE of lowest cost among the distinguishing ones.

Notice that most off-the-shelf planners (such as the MetricFF planner (Hoffmann, 2002), which we used in our experiments) do not guarantee that they actually find an optimal plan for efficiency reasons, but in practice the plans that our planner finds are close to optimal (see Section 6).

Object relations	
RoomSameTypeDisNum	the number of distractors of the same type as the referent in the room
MicroLandmarkInRoom	whether there are any micro-level (i.e. movable) landmarks in the room
MacroLandmarkNearby	whether there are any macro-level (i.e. immovable) landmarks near the referent
Spatio-visual	
Distance	the Euclidean distance (in GIVE space units) between the IF and the referent
Angle	the angle (in radians) between the center of the IF's field of view and the referent
Referent's distinctiveness	
ColorUnique	whether the referent's color is unique (i.e. not shared by other objects) in the world
LandmarkTypeUnique	whether a landmark with unique type in the world exists in the referent's room
Interaction history	
Round	the number of times the referent has been target of manipulation in a whole session
ReferenceAttempt	the number of times the referent has been referred to in the same round
SeenDeltaTime	the time elapsed (in seconds) since the referent was last seen by the IF

Table 2: Features putting the REs of the corpus into context.

5.1 Computing the costs of RE attributes

Each attribute that we might want to use as part of an RE is represented as a single planning operator in the planning problem of Section 3. The key problem we must solve is to determine the cost we want to assign to each of these operators.

We can approach this problem by inspecting how the individual attribute weights $v_j(s)$ contribute to the successfulness probability in (2). If for a given j, $v_j(s)$ is a negative value, then an RE r for which $a_j(r) = 1$ will have a higher $P(succ(r) = 1 \mid r, s)$ than an RE r' that is like r except that $a_j(r') = 0$. If $v_j(s)$ is positive, then the effect is reversed: choosing a_j will lower the probability of high successfulness. The effect that choosing a_j has on the probability grows with the absolute value of $v_j(s)$.

It therefore seems natural to use $v_j(s)$ as the cost of all planning operators for attributes of type a_j. Indeed, it can be shown that under this assumption, if a plan expresses the RE r, then the plan has minimal cost among all correct plans just in case r has maximal successfulness probability among all uniquely referring REs. Therefore we can reduce the problem of computing a successful RE to that of solving a metric planning problem.

5.2 Working around planner limitations

There is one final technical complication which we must address: Most off-the-shelf metric planners do not accept negative operator costs (because otherwise the action could be executed again and again in order to lower the total plan cost), but $v_j(s)$ may be a negative value. Such negative weight attributes improve the successfulness estimate of an RE even if they are not necessary to distinguish the referent, and we would like the NLG system to include them in the (redundant) RE it generates.

We work around this problem by introducing, for each attribute type a_j, a special action **non-**a_j. Executing this action in a plan corresponds to the choice to *not* include any attribute of type a_j in the RE; because it does not encode a lexicon entry from the TAG grammar, the action has no preconditions or effects pertaining to syntax or semantics. We can enforce that every RE must contain for every j either an attribute of type a_j or the action **non-**a_j by inserting atoms $\mathsf{needtodecide}(a_j, u)$ whenever some planning action introduces the RE u, and requiring that the final state of the planning problem may not include any needtodecide atoms. These atoms can be removed only by executing actions for attributes of type a_j or the action **non-**a_j. Now we assign the cost $\mathsf{cost}(a_j) = \max\{0, v_j(s)\}$ to each attribute action and the cost $\mathsf{cost}(\neg a_j) = \max\{0, -v_j(s)\}$ to **non-**a_j. Notice that $\mathsf{cost}(a_j) - \mathsf{cost}(\neg a_j) = v_j(s)$ regardless of whether $v_j(s)$ is positive or negative. Thus we obtain a metric planning problem in which all action costs are zero or positive, and whose minimal-cost plans correspond to maximal-probability REs.

5.3 An example

As an example, consider the planning operators for the attribute "red" and for **non-absolute**, shown in

red(u, x):
Precond: referent(x), canadjoin$(\mathrm{N}, u), \ldots$
Effect: $\neg$needtodecide(absolute$, u), \ldots$
Cost: cost(absolute)

non-absolute(u):
Precond: needtodecide(absolute$, u$)
Effect: $\neg$needtodecide(absolute$, u$)
Cost: cost($\neg$absolute)

Figure 4: Simplified mSCRISP planning operators for an absolute attribute.

Figure 4. These replace the operator for **red** shown in Figure 2; the other operators from Figure 2 are changed analogously.

The initial state of the planner might contain the atoms subst(NP, n_1) and ref(n_1, b) indicating that we want to generate an NP (with node name n_1 in the derivation tree) referring to b. Let's say it also contains the atoms button(b) and red(b), indicating that b is a red button. Lastly, there will be an atom needtodecide(absolute$, n_1$). The planner can start by selecting the action **the-button**(n_1, b), incurring the cost for a taxonomic attribute. The planner must then apply either the action **red**(n_1, b), incurring the cost for an absolute attribute, or the action **non-absolute**(n_1), with the cost of not choosing an absolute attribute; one of the two must be applied because we cannot be in a final state before all needtodecide atoms have been removed. If b is the only button in the domain, the choice between the two actions depends on which of cost(absolute) and cost($\neg$absolute) is greater. If another button exists, it may be that the planner is forced to apply **red** in order to distinguish b, regardless of the relative costs. In this way, the metric planner will not compute the cheapest combination of arbitrary attributes, but the cheapest RE among all uniquely referring ones.

6 Evaluation

We evaluate our model against two baselines. The MaxEnt baseline builds an RE by selecting all attributes a_j for which $v_j(s) \leq 0$ for a given scene s. This is a purely statistical model, which does not verify the applicability or discriminatory power of the attributes it selects, and thus makes no correctness or uniqueness guarantees. The EqualCosts baseline is a version of our mSCRISP model in which all attribute costs are equal. This is a purely symbolic model which always computes a correct and unique RE, but does this without any empirical guidance about expected successfulness.

Human	*the green button on the left*
MaxEnt	*the button to the left of the picture*
EqualCosts	*the left button,* *to the left of the right button*
mSCRISP	*the button to the left of the picture*

Table 3: REs produced by a human IG, our model and the two baselines in the bottom-left room of Figure 3.

Table 3 presents example REs that a human IG, our model and the two baselines issue for one of the buttons in the bottom-left room of Figure 3. As the IF is entering the room, they see from left to right a green button, a picture, and another green button. All REs in this example are distinguishing. However, the human-produced RE, which favors the use of an absolute ("green") and a viewer-centered ("on the left") attribute over one pointing to the micro-level landmark ("to the left of the picture"), was not particularly successful in the scene: After hearing it, the IF spent time scanning the room further to the left before finally approaching the referent. MaxEnt and mSCRISP generate a different RE, using a landmark, which they judge to be more successful. By contrast, EqualCosts generates a correct but more complex RE.

6.1 Accuracy of successfulness estimations

We train the maxent model on a dataset consisting of REs in the virtual worlds 1 and 2 of the GIVE-2 corpus. All evaluations are performed on a test set consisting of REs in world 3 (Figure 3). Both corpora contain all REs (a) in which the IF is already in the same room as the referent (so as to prevent interference between navigation instructions and REs) and (b) which only contain the attribute types shown in Table 1. This amounts to 358 REs in the training set and 174 REs in the test set.

The *accuracy* of the maxent model, i.e. the proportion of REs whose binarized successfulness it estimates correctly, differs between the training and test set. On the training data, the accuracy is 75.1%; on the test data, it is 62.1%. This compares favorably to a majority classifier, which would achieve

	succ. prob.
Human	0.467***
MaxEnt	**0.984****
EqualCosts	0.649***
mSCRISP	0.957

Table 4: Average probabilities of high successfulness. Differences to mSCRISP are significant at $^{**}p < 0.01$, $^{***}p < 0.001$ (paired t-tests).

	DICE		
	low succ.	high succ.	all
MaxEnt	0.320***	0.449*	0.371***
EqualCosts	**0.512**	0.475	**0.497**
mSCRISP	0.457	**0.519**	0.482
#REs	78	51	129

Table 5: Average DICE coefficients across datasets. Differences to mSCRISP are significant at $^{*}p < 0.05$, $^{***}p < 0.001$ (paired t-tests).

50% accuracy on the training dataset (since it is balanced); that is, the maxent model actually does learn to predict successfulness. The difference in accuracy shows that the training and test data are varied enough for a fair evaluation. In addition, the drop suggests that more training data might further improve the system's overall performance.

6.2 Successfulness probability

We now use our system and the two baselines to generate REs for the referents in the test corpus, and use the maxent model to estimate the probability (2) that the generated RE is in the high successfulness class. We define the domain entity set of the planning-based models to be the objects that are visible within the target referent's room, and we restrict ourselves to those scenes in which the target is among these objects. The results are shown in Table 4.

We find that the MaxEnt baseline significantly outperforms all other models. This is not surprising, as the metric of evaluation here is exactly what this baseline directly optimizes for. However, MaxEnt picks the different attributes independently, ignoring whether the resulting RE is semantically informative; correctness and uniqueness of an RE are not captured by the maxent model. Of the models which guarantee that the generated RE refers uniquely, mSCRISP performs the best.

6.3 Humanlikeness

Although this was not the main focus of this work, we also looked at the similarity of the system-generated REs with the original REs produced by the IGs. We model the degree of humanlikeness by the Dice coefficients of the two REs (Dice, 1945; Gatt et al., 2007). The results are shown in Table 5, both for all REs in the test set and for the REs of high and low human-achieved successfulness separately.

This test reveals that the REs computed by MaxEnt are less humanlike than those computed by either of the planning-based systems. This can be explained by the fact that, in contrast to MaxEnt, the planning-based models generate their REs on the basis of a set of correctness and uniqueness principles, which are, at least to some extent, shared by humans. Even though the difference is not statistically significant, mSCRISP reaches a higher degree of humanlikeness than EqualCosts on REs of *high* successfulness. Importantly, this is reversed in the low successfulness dataset. The distinction is relevant because mSCRISP does not attempt to mimic human IG choices under all circumstances; it only does so when it believes that the human IG choices are highly successful. If this is not the case, it makes different choices—those that a more successful IG might make in the situation.

6.4 Task-based evaluation

To verify the model's performance in the context of real interactions with human IFs, we entered mSCRISP and the correct RE generating baseline EqualCosts as participating NLG systems for the 2011 edition of the GIVE Challenge (Garoufi and Koller, 2011; Striegnitz et al., 2011). Both systems operate by first generating an RE (the *first-attempt* RE) for a given button target as soon as the IF is in the target's room and can see the target. Subsequently, the systems issue follow-up REs at regular intervals until the IF responds with a manipulation act or navigates away from the target.

Follow-up REs may differ from first-attempt REs, especially for the mSCRISP system, which relies for its attribute selection on several dynamically changing context features of the scene (see Table 2). Indeed, mSCRISP issues follow-up REs that are dif-

	resol. success		successfulness	
	all	non-rephr.	all	non-rephr.
EqualCosts	86%***	86%	0.32	0.38***
mSCRISP	**95%**	**89%**	**0.33**	**0.52**

Table 6: Task-based evaluation results. Differences to mSCRISP are significant at $***p < 0.001$ (Pearson's χ^2 test for resolution success rates; unpaired two-sample t-tests for the rest).

ferent from the original more often than the purely symbolic system (in 85% of the cases, as compared to only 59% for EqualCosts). Follow-up REs are important for the GIVE task, yet the fact that they are issued regardless of whether the IF is on the right track or not poses a problem on automatic methods of assessing success. We therefore base our analysis only on first-attempt REs. To control for the effect of rephrasing, we separately examine the subset of REs for which all follow-up REs were *non-rephrasing*, i.e. exactly the same as the original. We conduct the analysis on the latest currently available snapshot of the challenge results, which contains 74 valid games for each of our two systems. We first look into two metrics for referential success, as shown in Table 6.

In terms of resolution success, which represents the rate of REs whose intended referents have been correctly identified by the hearer (regardless of how fast), we find that mSCRISP significantly outperforms the baseline with a high success rate of 95%. Though the results are measured on different datasets and are thus not directly comparable, it is interesting to note that this surpasses the 92% success rate of human IGs in the GIVE-2 corpus. The system's performance remains better than the baseline's, though not significantly so, in the non-rephrased RE dataset. Turning to the metric of successfulness as defined in Subsection 4.2, we see that the two systems do not differ significantly when all first-attempt REs are considered. However it is clear that rephrasing affects the hearer's response, since processing new REs takes additional time. Examining the portion of non-rephrased first-attempt REs, we find that our model does generate REs that humans resolve significantly faster.

Finally, from the questionnaire data collected in the challenge, we consider a subjective metric of RE success as reported by the IFs in response to the post-task question "I could easily identify the buttons the system described to me". Although a Tukey's test does not find the difference to be statistically significant, it is worth mentioning that our model receives higher rates than the baseline with respect to this subjective metric, too. The average scores for mSCRISP and EqualCosts are 38.59 and 16.42, respectively (on a scale of -100 to 100).

7 Conclusion

In this paper, we have shown how to extend a symbolic system for generating REs with a statistical model of successful REs. Our system operates by training a maximum entropy model on a corpus in which the successfulness of REs is marked up, and mapping the maxent weights to action costs in a metric planning problem. Our evaluation, which also draws from real interactions with human hearers in the task-based setting of the GIVE-2.5 Challenge, shows that our model learns to distinguish highly successful attribute choices from less successful ones, and outperforms both a purely symbolic and a purely statistical baseline.

Although the system as we have presented it here builds on a planning-based model, nothing particular hinges on this choice: As far as generation of noun phrase REs is concerned, the planner makes similar choices to e.g. the system of Krahmer et al. (2003), and our cost function could be used in other systems as well. However, one strength of planning-based systems is that they are not limited to generating isolated noun phrases. In a situated setting like GIVE, it has been shown that they can be made to generate navigation instructions which (if successful) modify the non-linguistic context in a way that makes simpler REs possible later (Garoufi and Koller, 2010). It is an interesting issue for future work to extend our successfulness model to navigation instructions, and obtain a system that deliberately interleaves navigation and RE generation in order to maximize overall communicative success.

Acknowledgments

We are thankful to Ivan Titov and Verena Rieser for fruitful discussions about the maxent model, and to our reviewers for their many thoughtful comments.

References

Douglas E. Appelt. 1985. *Planning English sentences*. Cambridge University Press, Cambridge, England.

Daniel Bauer and Alexander Koller. 2010. Sentence generation as planning with probabilistic LTAG. In *Proceedings of the 10th International Workshop on Tree Adjoining Grammar and Related Formalisms*, New Haven, CT.

Anja Belz and Albert Gatt. 2007. The attribute selection for GRE challenge: Overview and evaluation results. In *Proceedings of UCNLG+MT*, Copenhagen, Denmark.

Anja Belz and Albert Gatt. 2008. Intrinsic vs. extrinsic evaluation measures for referring expression generation. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies*, Columbus, OH.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19.

Nina Dethlefs, Heriberto Cuayáhuitl, and Jette Viethen. 2011. Optimising natural language generation decision making for situated dialogue. In *Proceedings of the 12th annual SIGdial Meeting on Discourse and Dialogue*, Portland, OR.

Lee Raymond Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Maria Fox and Derek Long. 2003. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Int. Res.*, 20:61–124.

Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 Corpus of Giving Instructions in Virtual Environments. In *Proceedings of the 7th Conference on International Language Resources and Evaluation*, Valletta, Malta.

Konstantina Garoufi and Alexander Koller. 2010. Automated planning for situated natural language generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.

Konstantina Garoufi and Alexander Koller. 2011. The Potsdam NLG systems at the GIVE-2.5 Challenge. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, Nancy, France.

Albert Gatt and Anja Belz. 2010. Introducing shared task evaluation to NLG: The TUNA shared task evaluation challenges. In E. Krahmer and M. Theune, editors, *Empirical methods in natural language generation*, volume 5790 of *LNCS*. Springer.

Albert Gatt, Ielka van der Sluis, and Kees van Deemter. 2007. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the 11th European Workshop on Natural Language Generation*, Schloss Dagstuhl, Germany.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1).

Jörg Hoffmann. 2002. Extending FF to numerical state variables. In *Proceedings of the 15th European Conference on Artificial Intelligence*, Lyon, France.

Srinivasan Janarthanam and Oliver Lemon. 2010. Learning to adapt to unknown users: Referring expression generation in spoken dialogue systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.

Aravind K. Joshi and Yves Schabes. 1997. Tree-Adjoining Grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–123.

Alexander Koller and Matthew Stone. 2007. Sentence generation as planning. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic.

Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. The First Challenge on Generating Instructions in Virtual Environments. In M. Theune and E. Krahmer, editors, *Empirical Methods in Natural Language Generation*, volume 5790 of *LNCS*, pages 337–361.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.

Ivandre Paraboni, Kees van Deemter, and Judith Masthoff. 2007. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33(2):229–254.

Laura Stoia, Darla M. Shockley, Donna K. Byron, and Eric Fosler-Lussier. 2006. Noun phrase generation for situated dialogs. In *Proceedings of the 4th International Natural Language Generation Conference*, Sydney, Australia.

Matthew Stone and Bonnie Webber. 1998. Textual economy through close coupling of syntax and semantics. In *Proceedings of the 9th International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Canada.

Matthew Stone, Christine Doran, Bonnie Webber, Tonia Bleam, and Martha Palmer. 2003. Microplanning with communicative intentions: The SPUD system. *Computational Intelligence*, 19(4):311–381.

Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariet Theune. 2011. Report on the Second Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2.5). In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France.

Jette Viethen and Robert Dale. 2008. The use of spatial relations in referring expression generation. In *Proceedings of the 5th International Natural Language Generation Conference*, Salt Fork, OH.

Jette Viethen, Robert Dale, Emiel Krahmer, Mariet Theune, and Pascal Touset. 2008. Controlling redundancy in referring expressions. In *Proceedings of the 6th International Language Resources and Evaluation Conference*, Marrakech, Morocco.