Generation of effective referring expressions in situated context

Konstantina Garoufi and Alexander Koller
Area of Excellence “Cognitive Sciences”
University of Potsdam
Karl-Liebknecht-Str. 24-25
D-14476 Potsdam
Germany

Running head: Effective reference in situated context

Correspondence should be addressed to Konstantina Garoufi. Phone: +49 331 977 2951. Fax: +49 331 977 2925. Email: garoufi@uni-potsdam.de.
Abstract

In task-oriented communication, references often need to be effective in their distinctive function, that is, help the hearer identify the referent correctly and as effortlessly as possible. However, it can be challenging for computational or empirical studies to capture referential effectiveness. Empirical findings indicate that human-produced references are not always optimally effective, and that their effectiveness may depend on different aspects of the situational context that can evolve dynamically over the course of an interaction. On this basis, we propose a computational model of effective reference generation which distinguishes speaker behavior according to its helpfulness to the hearer in a certain situation, and explicitly aims at modeling highly helpful speaker behavior rather than speaker behavior invariably. Our model, which extends the planning-based paradigm of sentence generation with a statistical account of effectiveness, can adapt to the situational context by making this distinction newly for each new reference. We find that the generated references resemble those of effective human speakers more closely than references of baseline models, and that they are resolved correctly more often than those of other models participating in a shared-task evaluation with human hearers. Finally, we argue that the model could serve as a methodological framework for computational and empirical research on referential effectiveness.
Introduction

In task-oriented communication, speakers frequently produce distinctive referring expressions. The primary purpose of such expressions is to help the hearer uniquely identify the referent; a distinctive referring expression is effective if the hearer resolves it to the intended referent correctly and, ideally, effortlessly. As a consequence, computational models of reference in task-oriented settings typically aim at generating referring expressions that are as effective as possible. However, it can be challenging to capture effectiveness, as this, by definition, involves fine-grained observations about how hearers process referring expressions. Furthermore, whether a given referring expression is effective depends on the surrounding linguistic and non-linguistic properties of the referential scene, i.e., the situated context. Because of the number and complexity of the different factors involved, empirical research findings about what types of referring expressions (e.g., overspecified or not) are optimally effective, and under what circumstances, can be hard to generalize.

An increasing number of computational and empirical studies has been concerned with modeling or analyzing referential effectiveness. Computational models have frequently approximated the problem by generating expressions that are as humanlike as possible, i.e., that are optimized for resembling those produced by human speakers in similar contexts (see e.g., Viethen (2011) for an overview). However, empirical findings are mixed as to the extent to which human-produced references are easy for hearers to understand (e.g., Keysar, Lin, and Barr (2003)). Another common approximation is to assume that one can generate effective references by considering attributes of referents for selection one by one, as specified by a fixed preference order (Dale and Reiter, 1995). Yet empirical studies provide evidence that referential preferences of speakers and hearers (supposing that hearers prefer easy-to-understand expressions) vary according to dynamically evolving aspects of a referential scene’s context (see, e.g., van Deemter, Gatt, van der Sluis, and Power (2012)). The problem of understanding the exact influence of different aspects of context on referential preferences, and modeling effective reference production under these influences, has remained unsolved.
In the present work, we aim at addressing this problem. We propose an alternative approach to reference generation, which draws a distinction between less helpful and more helpful human behavior, and is explicitly concerned with modeling the latter. We are able to make this distinction by using an interaction corpus in a 3D environment, which records the hearers’ reactions along with the speakers’ referring expressions. On this corpus we train a maximum entropy model of the helpfulness of each attribute of a referring expression, given a certain scene. We then design and implement a reference generation model, mSCRISP, that incorporates the derived statistical estimations into the problem-solving technique of automated planning, to compute, for each referential context, a distinguishing referring expression of optimal estimated effectiveness. We find that mSCRISP manages to serve the needs of hearers well, while generating references that resemble those produced by effective human speakers. The model outperforms baseline models on referential effectiveness, in both automatic and human task-based evaluation. These results allow us to argue that, because mSCRISP is able to optimize effectiveness under selected, explicitly formalized aspects of situated context, it can serve as a methodological framework for computational and empirical research on referential effectiveness.

In the remainder of this article, we first review state-of-the-art computational models of reference and discuss them in the light of empirical findings about referential effectiveness. We then introduce the planning-based approach to sentence generation, which enables the generation of semantically valid references, and illustrate how we are able to rank these references according to their effectiveness by obtaining a statistical account of context-dependent attribute preferences. We go on to show how we combine these two types of reasoning to derive our model mSCRISP. Finally, we evaluate the model and discuss possible improvements and implications for future computational as well as empirical research.
Referential effectiveness: Computational models and empirical insights

The objectives of a computational model of reference are subject to the nature of the generation task at hand. In news or narrative discourse generation, for example, in which descriptive reference plays a major role (Hervas and Finlayson, 2010), it might be important to explore the breadth of human creativity in producing descriptions whose functionality goes beyond activating referents (Maes, Arts, and Noordman, 2004). In procedural discourse (Longacre, 1983), on the other hand, in which the speaker tells the hearer how to accomplish a given task, references primarily have the distinctive function of helping the hearer identify entities involved in the task. To serve this function, models need to generate effective referring expressions; otherwise an expression would be of small use to a hearer, regardless of how natural or otherwise fluent it might sound. In this section we examine, in the light of insights gained from empirical research, two main ways in which state-of-the-art computational models have typically approached this problem: optimizing humanlikeness and using fixed attribute preference orders.

The effectiveness of humanlike references

Computational models. State-of-the-art computational approaches to distinctive reference often aspire to generate referring expressions that are as humanlike as possible. For instance, Viethen, Dale, Krahmer, Theune, and Touset (2008) tune the parameters of the graph-based algorithm of Krahmer, Erk, and Verleg (2003) by computing attribute costs from the TUNA corpus (Gatt, van der Sluis, and van Deemter, 2007) in order to model the redundancy often found in human-produced references. Other approaches apply machine learning to human-produced data with richer representations of the situational context in their domains, with the purpose of varying their output in ways similar to human speakers (e.g., Jordan and Walker (2005); Stoia, Shockley, Byron, and Fosler-Lussier (2006)). In general, this line of research primarily attempts to replicate the referring expressions produced by humans, under the assumption that
human-produced references are also, for the large part, effective (Viethen, 2011).

**Empirical insights.** This is by no means an unfounded assumption; several studies have shown that human reference production is often hearer-oriented and specially designed to facilitate the identification process of the hearer. Particularly in interactive dialog settings, speakers and hearers have been observed to systematically collaborate towards establishing mutually acceptable forms of reference with an aim of minimizing their joint effort (Clark and Wilkes-Gibbs, 1986). References thus often become partner-specific, making identification easy for their particular addressee but not so much so for an overhearer or a new hearer (e.g., Schober and Clark (1989); Brown-Schmidt (2009)). Such audience-design mechanisms have been argued to be strong and early-onsetting (e.g., Brennan and Hanna (2009)). Even in non-interactive settings, common characteristics of human-produced referring expressions such as overspecification have been found to speed up identification (e.g., Arts, Maes, Noordman, and Jansen (2011)).

Another large body of research, however, provides conflicting evidence. Keysar et al. (2003), for instance, suggest that interlocutors sometimes fail to take the conceptual perspective of their partner into account during procedural interaction. Wardlow Lane and Ferreira (2008) find that speaker-internal cognitive pressures can be so powerful that they may override speaker-external communicative pressures, even when that threatens referential success. In the TUNA shared task on referring expression generation, Gatt, Belz, and Kow (2009) evaluate both human-produced and system-generated expressions using automatic measures of humanlikeness, human judgments of adequacy and fluency, as well as the referential clarity measures of accuracy and identification speed. The results provide compelling evidence that human-produced referring expressions are not necessarily effective: Measures of humanlikeness and referential clarity are not found to correlate in any significant way; in fact, human-produced referring expressions are systematically and significantly outperformed in terms of identification speed by the expressions that some of the systems generate for the task.
Conclusions. We conclude that both the speaker’s beliefs about the hearer’s ease of comprehension and the speaker’s own ease of production can influence human reference production. Factors such as the speaker’s cognitive load, the extent to which considerations of the hearer are salient in the interaction, as well as the severity of the consequences of being unclear, are all likely to play an important role in determining how the tension between speaker- and hearer-oriented processes will be resolved (Roßnagel, 2000; Haywood, Pickering, and Branigan, 2005). However, quantifying the exact influence of these and any other relevant factors is still a matter of ongoing experimental work. Optimizing human likeness therefore does not necessarily guarantee optimal effectiveness. To overcome this problem, computational models can directly assess human-produced references for their effectiveness and aim at reproducing only the ones among them that are effective. This is the approach we explore in this work.

Some recent computational works share this view and make explicit attempts to tailor models’ outputs to hearers’ needs. Paraboni, van Deemter, and Masthoff (2007), for example, present rule-based models that can deliberately generate redundant expressions in order to make referents in hierarchically structured domains easy to identify. The models can generate e.g. the redundant expression “the library in room 120 in Cockcroft” instead of the likewise distinguishing but less useful “the library”, as a means of helping a hearer locate the library of a university campus for the first time. Similarly, Guhe (2009) presents a model that decides upon the inclusion of color as an attribute of a referring expression according to the probability that the hearer knows the referent’s color. Golland, Liang, and Klein (2010) present a model of a “rational speaker”, which is based on a maximum entropy learner and generates references optimally with respect to an embedded hearer model. Also reinforcement learning techniques have been used to adapt to (human or simulated) users and optimize task success (e.g., Janarthanam and Lemon (2010); Dethlefs, Cuayáhuital, and Viethen (2011)). Nonetheless, most of these approaches have not yet been tried in tasks in which good content determination choices are less obvious, addressing problems of a broader scope is required, or realistic interactions with human hearers are involved. In this work, we address the problem of effective reference generation in complex situated context, and
Effective reference in situated context

test the performance of our approach in a shared-task human evaluation.

**The effectiveness of fixed preferences**

**Computational models.** Another approach commonly followed by computational models of reference for tackling the attribute selection task is the use of fixed preferences for processing attributes. This approach is motivated by the assumption that human speakers consistently prefer the use of certain forms of referring expressions over others, depending on factors such as the cognitive load of the speakers themselves or their hearers while processing these expressions. Indeed, speakers often prefer for example to include perceptually salient attributes of referents (such as color) in their expressions, even when this results in overspecified utterances (Pechmann, 1989). Such observations have been taken as evidence that each domain of reference may have its own, fixed, attribute preference order, based on which computational models should consider attributes for inclusion in a referring expression (e.g., Dale and Reiter (1995); Kelleher and Kruijff (2006); Gatt et al. (2007); Viethen et al. (2008)).

**Empirical insights.** However, psycholinguistic studies increasingly suggest that referential choices speakers make are not fixed throughout an interaction. Fukumura, van Gompel, and Pickering (2010), for instance, show that linguistic and non-linguistic context bears upon the choice of a pronoun over a repeated noun phrase when speakers refer back to a referent in a preceding utterance. The influence of the situational context on referential choice is not restricted to pronouns: Goudbeek and Krahmer (2010) find that attribute choice and modifier ordering are subject to priming and adaptation mechanisms arising in an interaction, while Koolen, Gatt, Goudbeek, and Krahmer (2011) further observe that attribute choice in both spoken and written human reference production is affected by features of the communicative setting as well as the referent. Also a corpus study by Viethen and Dale (2008) concludes that factors such as the salience of potential landmarks in a scene can encourage speakers to use e.g. more spatial relations in their referring expressions.

Not only speakers’ but also hearers’ preferences (supposing that hearers prefer
easy-to-understand referring expressions) seem to be sensitive to the situational context. For example, even though overspecification in referring expressions can under certain circumstances—especially when it allows the hearer to create a mental image of the referent or limit their search process—speed up identification (Arts et al., 2011), in other cases it can hinder it. In particular, the study of Engelhardt, Demiral, and Ferreira (2011) shows that unnecessary attributes in referring expressions can actually impair comprehension in simple visual scenes, even when they are realized in a syntactically unambiguous way. This negative effect arises also from the color attribute, which may come as a surprise; color has typically been considered a highly preferable attribute, whose use, even when unnecessary, is favored by speakers and hearers alike (e.g., Dale and Reiter (1995)). In any case, it remains an open issue how such results would generalize in more complex referential scenes with richer visual and other types of context.

Conclusions. All considered, modeling referential preferences as fixed and uniform for all situations in a referential domain may not necessarily lead to optimally effective expressions, as it does not seem to reflect mechanisms of either human-made choices or choices that result in easy-to-understand output. Instead, computational models of reference can increase their effectiveness by making their choices sensitive to dynamic aspects of a scene’s context. For instance, as van Deemter et al. (2012) argue, preference for the color attribute in a visual scene should be subject to the degree of its perceptibility, among other factors, which in turn might depend on how far in the given scene referents are located. Further potentially important factors that van Deemter et al. (2012) draw attention to include the discriminatory power and the “extremity” of an attribute in the scene, intentional influences, as well as the dynamically evolving mechanisms of alignment. Although identifying the relevant aspects of linguistic and non-linguistic context that come into play and measuring their effect is certainly not a trivial task in complex scenes, we show with our model how such a task could be approached.
Planning referring expressions

Our model builds upon CRISP (Koller and Stone, 2007), an approach to natural language generation which handles the full sentence-generation problem as an automated-planning problem. Although we only use this approach to generate individual noun phrases here, these are in fact part of an expressive integrated sentence planning and realization process, which has also been extended to the generation of entire discourses (Garoufi and Koller, 2010). Generation with CRISP involves the following two main stages: converting a language-generation problem into an automated-planning problem, and providing a solution to the former by solving the latter.

Converting language generation problems into planning problems

As a source of linguistic knowledge about the expressions it generates, CRISP utilizes a lexicalized tree adjoining grammar (Joshi and Schabes, 1997). In this formalism, each lexical item is associated with an elementary tree encoding a certain phrasal structure, as in Figure 1. Such trees can be combined with each other by means of substitution and adjunction operations, as specified by the grammar. This way increasingly larger trees are derived, which correspond to sentence constituents and, ultimately, full sentences. To allow for the generation of meaningful expressions, we enrich the lexicon with semantic and pragmatic information in addition to the syntactic information it encodes. CRISP 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 referential scene. Given an example knowledge base \{button\left(b_1\right), \text{red}\left(b_1\right), \text{button}\left(b_2\right), \text{blue}\left(b_2\right), \text{left}_{\text{of}}\left(b_2, b_1\right), \text{chair}\left(c_1\right)\}, and a communicative goal that requires describing $b_1$, Figure 1 shows with a simplified version of the lexicon how the grammar derives the expression “the red button” referring to $b_1$.

(Figure 1 about here)

In order to arrive at the generation of this expression, CRISP converts the lexicon of
Figure 1 and the given communicative goal into an automated-planning problem (Ghallab, Nau, and Traverso, 2004), which is the problem of finding a sequence of operators whose execution will achieve the specified goal. In this conversion, the entries of the lexicon are encoded as individual planning operators. The preconditions of an operator determine which logical propositions must be true in a given planning state so that the operator can be executed, while its effects specify how the truth conditions of these propositions will change after the execution. The operators integrate linguistic and non-linguistic preconditions and effects, as shown in simplified form in Figure 2; in particular, the operators red and left, which encode a context-independent and a context-dependent attribute, respectively, include preconditions determining the eligibility of an entity to be described as “red” or “left” at a given state of the derivation (Garoufi and Koller, 2010). 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 with the satisfaction of a set of constraints. Syntactic constraints postulate syntactic completeness, while semantic constraints require that any entity referred to can be distinguished from other entities, called distractors, thus making sure that the generated referring expressions are distinguishing.

(Figure 2 about here)

Solving planning problems to generate language

This conversion makes it possible to generate referring expressions by reasoning about how the available lexical entries can be organized into correct and distinguishing descriptions of the referents, as encoded formally in the planning problem. CRISP outsources this task to an off-the-shelf dedicated planning system, which allows it to benefit from the efficiency of modern planning algorithms. Let us examine step by step what reasoning a planning system may follow for the generation of an expression describing $b_1$, given the example knowledge base we specified and the operators of Figure 2. The system can rule out as distractors for $b_1$ any entities that are not buttons, by executing the operator the–button applied to an available syntax node $n_1$ and the
entity \( b_1 \), i.e. the action \texttt{the-button}(n_1, b_1). This rules out \( c_1 \), since it is a chair, but the second button of the knowledge base \( b_2 \) remains as a distractor. Because it has a goal of eliminating all distractors, the system goes on to check the preconditions of other potential actions. It finds that \texttt{left}(n_1, b_1) is not applicable, as the knowledge base stipulates that \( b_2 \) is the left one among the two buttons, and this entails that \( b_1 \) fails to satisfy the action’s preconditions. However, action \texttt{red}(n_1, b_1) is applicable, as \( b_1 \) is red. Since this action now eliminates \( b_2 \) (which is blue) as a distractor, and the noun phrase is syntactically complete, the planner has achieved its goal and can terminate.

CRISP finally realizes the computed plan, i.e., the goal-reaching sequence of actions found, as the referring expression “the red button”.

A statistical account of referential effectiveness

Though the symbolic reasoning of CRISP guarantees the generation of semantically valid and distinguishing referring expressions (given a correct and complete model of the referential scene), these expressions are not necessarily optimal with respect to their effectiveness. In this section, we explain how we obtain a statistical account of effective referential choices in situated context. We start off with a human-human interaction corpus, in which we analyze hearers’ reactions after being presented with referring expressions. This enables us to distinguish between more effective and less effective references. We also model contextual properties of the referential scenes in order to capture the effects of different aspects of context on referential effectiveness. The resulting annotations allow us to train a maximum entropy learner that can predict, for new referential contexts, whether a reference will likely be effective or not.

Situated reference in the GIVE-2 corpus

We use the GIVE-2 corpus of Giving Instructions in Virtual Environments (Gargett, Garoufi, Koller, and Striegnitz, 2010). In this corpus, a human instruction giver (IG) types text to guide a human instruction follower (IF) through a virtual 3D world, with the goal of completing a treasure-hunting task. The worlds comprise entities of dif-
different types (e.g., movable objects such as furniture pieces and immovable features of rooms such as doorways), including buttons on walls, which IFs can approach and press. Figure 3 presents a top-down view of one of the three corpus worlds.

(Figure 3 about here)

During the interactions, IGs refer to a series of target buttons, which are buttons that IFs need to press in order to progress in the task. Gargett et al. (2010) have created an annotation scheme for these referring expressions, which classifies them according to the basic types of attributes of which they are made up, as shown in Table 1. Applying this scheme to the full English edition of the corpus, we find that human IGs describe target buttons most frequently in terms of their color (which is the only absolute attribute used), type, and spatial relations with respect to the hearer, landmarks or button distractors in the scene. In this work, we focus on the six most frequent attribute types, as illustrated in the upper six entries of Table 1. Of the 714 referring expressions annotated, 598 only use attributes of these types.

(Table 1 about here)

**Measuring effectiveness**

In this task-based setting, we can assess whether a referring expression to a target button has served its purpose by examining whether it led the IF to press the intended referent. A manual annotation, based on this examination, reveals that 92% of all expressions referring to target buttons in the corpus allow the IF to correctly identify the referent (regardless of how long it takes them). We can achieve a more even split of the data by assuming that an IF who understands the expression easily will walk towards the correct referent quickly and directly; in other words, the average speed at which they approach the referent is an indication of effectiveness. We thus define the measure of \( \text{successfulness} \) \( \text{succ}(r) \) of a referring expression \( r \), which is intended to
model computationally the linguistic property of referential effectiveness, as follows:

\[
\text{succ}(r) = \begin{cases} 
0 & \text{if } r \text{ was not correctly resolved} \\
\frac{\Delta S}{\Delta T} & \text{otherwise}
\end{cases}
\] (1)

where \(\Delta S\) is the distance in the GIVE world (including turning distance) between the target button and the IF’s location at the time when they encounter the expression \(r\), and \(\Delta T\) is the time elapsed between encountering \(r\) and pressing the referent. We can now split the referring expressions into a class of high successfulness and one of low successfulness, as follows:

\[
\text{succ}^*(r) = \begin{cases} 
0 & \text{if } \text{succ}(r) \leq \tilde{S} \\
1 & \text{otherwise}
\end{cases}
\] (2)

where \(\tilde{S}\) is the median of all values that \(\text{succ}(r)\) takes for all referring expressions \(r\) in the data. This binarized successfulness abstracts away from the exact numeric value of an expression’s successfulness, which is not important for our purpose, and allows us to create a balanced dataset with two classes of equal size. We examine this modeling choice further in the discussion section of the article.

**Modeling the situated context of referential scenes**

Referents (and distractors) in our corpus are situated in particular spatio-visual configurations and are associated with certain properties (e.g., discourse history). We call such sets of objects, with any properties they have, referential scenes. The surrounding linguistic and non-linguistic properties of referential scenes characterize their context. We formalize this notion of situated context via a collection of context variables, which represent individual measurable properties of scenes. Though variables can be defined to capture other possibly important influences of context on referential choices, such as the codability of the referent’s attributes in a scene (Viethen, Goudbeek, and Krahmer, 2012), we focus on the hearer’s conceptual accessibility of the referent here (e.g., Fukumura et al. (2010); Arts et al. (2011)). Table 2 presents the basic set of ten vari-
ables we define on these grounds.

We compute the values of these variables from the corpus automatically, except for the \textit{Round} and \textit{ReferenceAttempt} variables, which we annotated manually. Variables in the \textbf{OBJECT RELATIONS} and \textbf{REFERENT’S DISTINCTIVENESS} groups view the referent in relation to other objects, and could be defined in terms of different scopes of comparison: For the \textit{ColorUnique} variable, for instance, one could ask whether the referent has unique color among the objects (a) near it, (b) in the room, or (c) anywhere in the virtual world. We choose as scope for these variables the one that yields best results during the training of our statistical model, as indicated in Table 2. Notice that the \textit{Angle} variable subsumes whether a referent is visible, which likely exerts strong influence on referential choices (Stoia et al., 2006).

\section*{A maximum entropy model of effectiveness in context}

Now we combine the information we collected about human referential choices, their relative successfulness, and the context in which they were made, in order to train a maximum entropy model that can estimate the successfulness of any referring expression in any context. We assume that in a given context, all attributes of the same type as classified in Table 1 are equally effective for a hearer. Based on this assumption (which we examine in the discussion), we model a referring expression $r$ as a set of attribute types, and let $a_j(r) = 1$ if $r$ contains an attribute of the $j$-th attribute type of Table 1 ($j = 1, \ldots, 6$). Otherwise, $a_j(r) = 0$. For a referential scene $s$, we let $c_i(s)$ take the value of the $i$-th context variable of Table 2 in this scene ($i = 1, \ldots, 10$). We then combine attribute types and context variables into features of the form:

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

(3)

These features allow us to cast the problem as a simple binary classification task, in
which our goal is to estimate the conditional probability of a referring expression $r$ presented in a scene $s$ being highly successful, given a joint representation of attribute types and context:

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

We train a maximum entropy model to learn this distribution; this will later allow us to convert the model’s parameters into parameters for planning in order to steer CRISP’s attribute choices towards high successfulness. Maximum entropy models for binary classification tasks (high/low successfulness) are equivalent to logistic regression, as implemented e.g. in the Weka data mining workbench (Hall et al., 2009), which we use here. The model estimates the above probability as

$$\hat{P}( succ^* (r) = 1 \mid \{ \phi_{ij}(r, s) \}_{i,j} ) = \frac{1}{e^{\sum_{i,j} (w_{ij} \cdot \phi_{ij}(r, s)) + w_0} + 1},$$

for coefficients $w_{ij}$ and intercept $w_0$. By letting

$$v_j(s) = \sum_i (w_{ij} \cdot c_i(s)),$$

we derive:

$$\hat{P}( succ^* (r) = 1 \mid r, s) = \frac{1}{e^{\sum_j (v_j(s) \cdot a_j(r)) + w_0} + 1}.$$

This way, we obtain a weight $v_j(s)$ for each attribute type $a_j(r)$ of a reference $r$ in a scene $s$. In our data, we observe that every context variable of Table 2 affects the weight of at least one attribute type.

**Optimizing effectiveness using metric planning**

The weight of an attribute type as defined in (6) provides an estimation of that attribute’s contribution to the successfulness of a certain reference in a scene, according to (7). In this section, we explain how we combine these statistical estimations with
CRISP’s symbolic reasoning about referential correctness. We first describe how we employ the formalism of metric planning to assign individual costs to CRISP’s planning operators, associating each attribute type with an estimation of how preferable it is in the given context. We then illustrate how we work around planner limitations to derive our model, mSCRISP.

Assigning costs to attributes

We employ metric planning (Fox and Long, 2003), which is a form of automated planning that can handle numeric reasoning. This allows us to assign to each planning operator a numeric cost, such that the use of the operator in a plan will add its cost to the total cost of the plan. We further introduce a plan metric, which specifies that a planner should try to find a plan of minimal total cost; as it plans referring expressions, the system thus evaluates them according to their quality (as determined by their total cost), and looks for optimal-quality ones. Though off-the-shelf planners may not guarantee that they actually find an optimal plan for efficiency reasons, in practice the plans that our planner Metric-FF (Hoffmann, 2002) finds are close to optimal (see evaluation results). This way we can reduce the problem of computing an effective referring expression to that of planning under an appropriate cost metric.

Following the CRISP model, we represent each attribute value that we might want to include in a referring expression as a single planning operator of a planning problem, as in Figure 2. The key problem we must now solve is to determine what cost to assign to each of these operators, so that the most preferable attribute choices receive the lowest costs. We can approach this by inspecting how the individual attribute weights \( v_j(s) \) contribute to the value of the probability in (7). If for some \( j \), \( v_j(s) \) is a negative value in scene \( s \), then \( \hat{P}(\text{succ}(r) = 1 \mid r, s) \) is higher for a reference \( r \) such that \( a_j(r) = 1 \), rather than if \( a_j(r) = 0 \). That is, choosing to include the attribute \( a_j \) in this case will increase the probability that the resulting reference will be highly successful. If \( v_j(s) \) is positive, then the effect is reversed: Choosing \( a_j \) will lower the probability of high successfulness. This means that, given a scene \( s \), we can determine the optimal
choice of attributes for a reference \( r \) by the attributes' weights in \( s \), as follows:

\[
a_j(r) = \begin{cases} 
1 & \text{if } v_j(s) \leq 0 \\
0 & \text{otherwise.}
\end{cases}
\]  

(8)

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 operators for attributes of type \( a_j \).

**Working around planner limitations**

We must address one final technical complication: Most off-the-shelf metric planners do not accept negative costs (because otherwise an action could be executed again and again in order to lower the total cost), but \( v_j(s) \) may be a negative value in a scene \( s \). Such negative-weight attributes improve the successfulness estimate of an expression even if they are not necessary to distinguish the referent, and we would like the generation model to include them in its (redundant) referring expressions.

We work around this problem by introducing, for each attribute type \( a_j \), a new operator \texttt{non-}a_j. This operator does not correspond to a lexical entry and lacks any preconditions or effects pertaining to syntax or semantics. Its presence in a plan represents a deliberate choice not to include any attribute of type \( a_j \) in a referring expression. To enforce that the planner will consider every available attribute while making its choices, we further introduce formulas \( \text{needtodecide}(a_j, u) \) for each attribute \( a_j \) and syntax node \( u \) that holds a referring expression. These formulas convey that the planner needs to decide whether or not to include \( a_j \) in an expression. We ensure this by setting an additional planning goal that no \( \text{needtodecide} \) formulas remain at the end of the planning process. Finally, we insert \( \neg \text{needtodecide} \) effects in such a way that removing these formulas is possible only by executing operators for attributes of type \( a_j \) or the operator \texttt{non-}a_j. This means that, to arrive at a valid referring expression, the planner must decide, for every attribute, whether to include it in the expression or not.

The planner makes these decisions based on the cost that each outcome incurs. To
favor good choices as dictated by (8), we assign the cost

$$cost(a_j) = \max\{0, v_j(s)\}$$

(9)

to each operator that represents an attribute of type $a_j$, and the cost

$$cost(\text{non-}a_j) = \max\{0, -v_j(s)\}$$

(10)

to the operator $\text{non-}a_j$. Notice that the cost of an attribute type depends on the referential scene $s$ (as seen through the context variables). We present an example cost assignment for our six attribute types in Table 3.

We thus obtain a metric planning problem in which all operator costs are positive or zero and whose minimal-cost plans correspond to maximal-probability referring expressions. Because the original planning problem (as constructed by CRISP) already enforces that a referring expression must be distinguishing, this amounts to finding the referring expression of lowest cost among the distinguishing ones.

Generating referring expressions with mSCRISP

As an example of how the resulting model mSCRISP operates, consider the planning operators for the attribute value “red” and for $\text{non-absolute}$, shown in Figure 4. These replace the operator for red shown in Figure 2; the other operators from Figure 2 change analogously.

(Figure 4 about here)

Let’s suppose we have a knowledge base that contains \{button($b$), red($b$)\}, stating that $b$ is a red button. The initial state of the planning problem might contain the formulas
Effective reference in situated context

subst(NP, n₁) and referent(n₁, b), indicating that we want to generate a noun phrase on syntax node n₁ that refers to b. Because n₁ holds a reference, there will also be formulas needtodecide(taxonomic, n₁) and needtodecide(absolute, n₁). (We ignore all other attributes for this example.) Now suppose that the planner executes the action the-button(n₁, b), deciding to include a taxonomic attribute and incurring its cost. This removes needtodecide(taxonomic, n₁) from the planning state. At this point, the planner must consider executing either the action red(n₁, b), incurring the cost for an absolute attribute, or the action non-absolute(n₁), with the cost of not choosing an absolute attribute. One of the two actions must be executed eventually, as it is impossible to arrive at a final state before all needtodecide formulas have been removed. In case b is the only button in the domain, the choice between the two actions depends on which of cost(absolute) and cost(non-absolute) is greater. If, however, a distractor exists and it is not red, the planner may be forced to apply red in order to distinguish b from that distractor—regardless of the relative costs. mSCRISP does not compute the cheapest combination of arbitrary attributes, but the cheapest combination among those that result in a distinguishing referring expression.

Automatic evaluation

To assess the adequacy of our approach, we evaluate mSCRISP with respect to intrinsic and extrinsic measures. In this section, we present an automatic evaluation study against a purely statistical and a purely symbolic baseline model in referential scenes of a GIVE-2 corpus world (shown in Figure 3). We find that our model generates more highly successful references than the purely symbolic baseline, according to the estimations of (7), and that its references are more similar to highly successful human-produced ones than those of either of the baselines.

Methods

The models. We design two baseline reference generation models to compare mSCRISP against. The MaxEnt baseline builds a reference in a scene s by select-
ing all attributes of type $a_j$ for which $v_j(s) \leq 0$, as prescribed by (8). That is, this baseline makes its choices exclusively based on the successfulness estimations of the maximum entropy model of (7), without combining those with reasoning about the semantics of the resulting references like mSCRISP does. This is therefore a purely statistical model, which does not verify the applicability or discriminatory power of the attribute types it selects, and thus makes no correctness or uniqueness guarantees.

The EqualCosts baseline, on the other hand, is a version of our mSCRISP model in which all attribute costs are equal. That is, unlike mSCRISP and the MaxEnt baseline, this baseline does not choose attributes by considering their contribution to successfulness according to (8). It is a purely symbolic model which computes correct and distinguishing referring expressions, but does this without any guidance about their expected successfulness. Finally, because we conduct this evaluation in referential scenes of a GIVE-2 corpus world, we also have human-produced references (Human) to compare the models’ choices against. Table 4 presents example referring expressions that these three different IGs produce for one of the buttons in the bottom-left room of Figure 3.

(Table 4 about here)

As the IF is entering the room, they see from left to right a green button, a picture, and another green button. All referring expressions in this example are distinguishing. However, the human-produced expression, 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 effective in the scene: After encountering it, the IF spent time scanning the room further to the left before finally approaching the referent. The MaxEnt baseline and mSCRISP generate a different expression, using a landmark, which they estimate to be more effective. By contrast, EqualCosts’s referring expression is correct but more complex.
**Procedure.** We train the maximum entropy model of (7) on a dataset consisting of referring expressions in the virtual worlds 1 and 2 of the GIVE-2 corpus. We then perform automatic evaluations on a test set consisting of expressions in world 3 (Figure 3). Specifically, we use mSCRISP and the two baselines to generate expressions for the referents in the test corpus, and estimate the probability that the generated references fall into the high successfulness class. We construct the knowledge bases of the planning-based generation models mSCRISP and EqualCosts to include the objects that are visible by the IF within the target referent’s room, and we restrict ourselves to those scenes in which the target is among these objects. Finally, to determine to what extent mSCRISP aligns effectiveness and humanlikeness, we look at the similarity of the generated references to those originally produced by the human IGs in the corpus. We model this similarity by the *Dice coefficient* metric (Dice, 1945; Gatt et al., 2007), which, examining references on the level of attribute selection (rather than lexicalization or surface realization), is in line with the focus of this work.

In both the training and the test set, we include only referential scenes in which (a) the referent is in the same room as the IF (so that it is visible by the IF or near them; this is meant to reduce the interference of navigation instructions), and (b) the referring expression only contains the attribute types shown in Table 1. This amounts to 358 referential scenes in the training set and 174 scenes in the test set.

**Results**

**Accuracy of successfulness estimations.** The *accuracy* of the maximum entropy classifier, i.e. the proportion of references in the given scenes whose binarized successfulness is estimated correctly according to (7), 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 50% accuracy on the training dataset (since it is balanced); that is, the maximum entropy model does learn to predict successfulness to some degree. The difference in accuracy indicates 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 mSCRISP’s overall performance.


**Probability of being highly successful.** Table 5 presents, for each model, the average probability (7) that the references it produces fall under the high successfulness class.

(Table 5 about here)

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 is designed to optimize for. However, MaxEnt picks the different attributes independently, ignoring whether the resulting expression is semantically informative; correctness and uniqueness of a referring expression are not captured by the statistical model. Of the reference generation models which warrant that the generated expression refers uniquely, mSCRISP performs the best.

**Humanlikeness.** Table 6 presents average Dice coefficient results, both for all references in the test set and for those of high and low human-achieved successfulness separately.

(Table 6 about here)

This test reveals that the expressions computed by the MaxEnt baseline are less humanlike than those computed by either of the planning-based generation models. This can be explained by the fact that, in contrast to MaxEnt, the planning-based models generate their expressions on the basis of a set of referential correctness and uniqueness principles, which are, at least to some extent, shared by humans. Though the difference is not statistically significant, mSCRISP reaches a higher degree of human-likeness than EqualCosts on references of high successfulness; 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 these choices are highly effective. If this is not the case, it makes different
choices—those that a more effective human IG might make.

**Human task performance evaluation**

The automatic evaluation results rely on the estimations of a statistical model, which may not be fully representative of the effectiveness of references in scenes with human hearers. To assess the performance of our model in the context of real interactions, we participated in the Challenge on Generating Instructions in Virtual Environments (GIVE-2.5; Striegnitz et al. (2011)). Systems participating in this shared task engaged in the role of generating written instructions to guide human IFs through a virtual treasure hunt. The virtual worlds were designed to be similar in nature to the GIVE-2 corpus worlds that were available for training, but also to provide reference generation models with challenges of varying complexity. We next present results of this human evaluation, focusing on the model’s referential performance. We find that mSCRISP’s references are resolved correctly more often and faster than those of our symbolic baseline, and lead to fewer errors on behalf of hearers than those of any other model participating in the shared task.

**Methods**

**The models.** We implemented mSCRISP and the purely symbolic baseline EqualCosts (which is the only one of our two baselines that always generates correct and distinguishing referring expressions) as parts of GIVE natural language generation systems. Both systems operate by first generating a first-attempt reference for a given target button as soon as the IF is in the target’s room and can see the target. Subsequently, they generate follow-up references at regular intervals until the IF responds by either pressing some button or navigating away from the target. Figure 5 shows an example of a reference situated in one of the GIVE evaluation scenes, as generated by mSCRISP. In this scene, EqualCosts would generate the different expression “the right one to the right of the green button”.

In addition to our own models, six other reference generation models participated in the GIVE-2.5 Challenge: Systems A, C, L and T generate references following hand-crafted approaches, while systems B and CL both base their referential choices on human production; CL selects individual references from a dedicated human-human interaction corpus, and B constructs references based mostly on a decision tree learnt from the GIVE-2 corpus. This latter system represents a supervised-learning approach applied to the same corpus as we use, which, however, optimizes references for humanlikeness rather than effectiveness.

**Procedure.** We first compare mSCRISP against EqualCosts with respect to two metrics of referential success: resolution success, which represents the rate of expressions whose intended referents have been correctly identified by the IF (regardless of how fast), and successfulness, as defined in (1). Then, we compare the model’s performance with the other models that were entered into the shared task evaluation. Though resolution success and successfulness rates are not immediately available for comparison, Striegnitz et al. (2011) report on similar measures that reflect to what degree IFs could identify the systems’ intended referents. In particular, the error rate of a system denotes the average number of incorrect button presses over the total number of actions performed in a single interaction. We must draw this comparison with caution, since the different approaches of systems to execution monitoring and repair may also bear on the prevention of errors. However, the mSCRISP system only uses simple execution monitoring techniques (see Garoufi and Koller (2011b)); and in any case, one may expect that if referring expressions are effective, misunderstandings are less likely to arise in the first place.

Both mSCRISP and EqualCosts generate follow-up references at regular intervals, which may differ from first-attempt ones. Such references are important for the GIVE task, yet the fact that they are generated regardless of whether the IF is on the right track or not poses a problem on assessing referential success. We therefore base our
analysis of resolution success and successfulness only on first-attempt references. To control for the effect of rephrasing, we separately examine the subset of references for which all follow-up references were non-rephrasing, i.e. exactly the same as the original. The results are derived from a total of 536 human-system interactions, which Striegnitz et al. (2011) collected over the Internet.

Results

Comparison against our own baseline. Table 7 presents average resolution success and successfulness rates for mSCRISP and the purely symbolic baseline.

(Table 7 about here)

In terms of resolution success, we find that mSCRISP significantly outperforms the baseline. Though the results are measured on different datasets and are thus not directly comparable, mSCRISP’s success rate of 95% 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 reference dataset. Turning to the metric of successfulness, the two systems do not differ significantly when all first-attempt references are considered. However, rephrasing may affect an IF’s response, since processing new expressions can take additional time. Examining the portion of non-rephrased first-attempt references, we find that the system does generate expressions that human IFs are able to resolve significantly faster.

Comparison against other reference generation models. Striegnitz et al. (2011) report upon error rates for all GIVE-2.5 participating systems as in Table 8.

(Table 8 about here)

We observe that, although pairwise Tukey’s tests do not find all differences to be statistically significant, mSCRISP outperforms the other systems with respect to this final
measure of referential performance. Further comparative results from Striegnitz et al. (2011) rank mSCRISP among the top systems on most objective and subjective evaluation measures, including overall duration and task success.

Discussion

The evaluation shows that mSCRISP can serve the needs of hearers well, while generating references that resemble those produced by effective human speakers. The model derives its ability to refer effectively both from its planning component, which ensures that references are correct and distinguishing, and from its corpus-based learning. In both components, we made a number of modeling choices and assumptions. In this section, we discuss these choices and how the model could be improved by exploring alternatives for these. Finally, although mSCRISP is primarily a computational approach to reference generation, it may support empirical studies into human reference processing. We end the discussion by examining some implications of this work for future computational and empirical research.

Improving the model

Our definition of successfulness in (1) as a metric that models referential effectiveness considers the time window from the presentation of a referring expression until the hearer’s action. This definition captures two important aspects of referential understanding: interpretation, which relates to determining the meaning of an utterance, and resolution, which involves identifying the referent, once a referring expression has been interpreted (Paraboni et al., 2007). On the other hand, this time window also includes the process of (simulated) physical interaction with the referent. The cognitive load of this subtask likely affects the hearer’s reaction speed and may also need to be factored in. Apart from that, processing a scene has an intrinsic cognitive load due to the inherent characteristics of that specific scene, which an improved metric of effectiveness would ideally account for. Another limitation is that it may be harder to detect referential understanding in domains without physical interaction. In certain situated
domains, this may be achieved using other observable cues, such as eye movements (Staudte, Koller, Garoufi, and Crocker, 2012). In any case, the precise quantification of referential effectiveness is a matter for further research.

The model chooses attributes assuming that in a given context, all attributes of the same type as classified in Table 1 are equally effective for a hearer. This grouping allows us to make good use of our limited training data, but is an oversimplification: For instance, color and shape are both absolute attributes, but their inclusion in a referring expression can result in significantly different identification times for hearers (Arts et al., 2011). Similarly, color may be more preferable in describing an elephant that is pink than in describing a gray one (van Deemter et al., 2012). Since the model assigns an individual cost to each lexical entry, it could be refined in such a way that every attribute (or even every value of an attribute) receives by the maximum-entropy learner its own cost. Given this kind of refinement, the mechanism of context variables may go a long way toward accounting for such context-dependent preferences (see, e.g., the \textit{ColorUnique} variable of Table 2). The set of context variables we selected here was intended as a starting point and could be extended to capture additional aspects of the linguistic and non-linguistic context.

Finally, an error analysis shows that the most problematic expressions the model generates are referring expressions with recursive structure such as “the button to the left of the right button”, “the button below the upper button”, and variants of these. These constructions arise primarily because our account of effectiveness focuses on attributes of the main referring expression and does not consider any embedded ones; e.g. it does not address the problem of optimizing the noun phrase complement “the right button” in “[the button to the left of [the right button]]”. It turns out that the base-line EqualCosts was more prone to this kind of expressions than mSCRISP, which at first glance could be hypothesized as a possible reason for the lower referential success rates of that model. However, examining the portion of non-rephrased expressions of each model that do not display this particular structure, we still find that mSCRISP’s references were significantly more effective. Despite this fact, extending the model to optimize embedded references could further improve its performance.
Implications for computational research

This work stands among other recent approaches that design referring expressions to suit the hearers’ needs. Because of the complexity of this problem, models so far have mostly focused on simpler tasks, e.g. whether or not to redundantly use the room-number and building-name attribute in a reference to a place that the hearer is not familiar with (Paraboni et al., 2007). By contrast, our approach makes it possible to optimize effectiveness in richer communicative settings, where a model has a higher number of non-trivial choices to make. Capturing effectiveness with a maximum entropy classifier is natural, in that maximum entropy models try to make minimal assumptions about the probability distribution beyond the empirical observations. Similar models have also been used to capture the process of an agent making a choice between discrete alternatives in other domains (Train, 2009). While we used it for a binary classification task here, it would be interesting for future computational work to employ this statistical model for a more fine-grained characterization of effectiveness.

The model is sensitive to diverse aspects of the situated context and can dynamically adjust its output to match the degree of saliency of different objects. Because it has access to information about the interaction history, it can also adapt its output interactively. As an example, the ReferenceAttempt context variable (Table 2) might be able to steer mSCRISP towards choosing a highly overspecified new reference, if it records that the model has unsuccessfully attempted to refer to a given referent before. Because it integrates sentence planning and realization, the model is not limited to content determination but can do full-fledged generation of references as parts of sentences. This could help generation models overcome the limitations of producing one-shot references in a null context and move towards references in which the surrounding linguistic (and non-linguistic) context also plays a role. As our approach is not domain-specific, it could transfer to other domains. Exploring grammar design and cost assignment for different domains would be interesting directions for future computational work.
Implications for empirical research

Our model functions in situated context, where spatio-visual and other non-linguistic context bears on referential preferences. Because of the interactive communicative setting we consider, it is possible to study reference as part of a longer interaction rather than as an isolated process. Empirical studies can thus be conducted in a more complex and realistic domain, which may be useful in generalizing the observations made in simple visual scenes (e.g., Engelhardt et al. (2011)). At the same time, such studies can often benefit from explicit formalization of the mechanisms involved (see e.g., Krahmer (2010)). With its computational modeling of the situated context through context variables, mSCRISP formalizes this notion.

From a hearer’s perspective, mSCRISP is based on a model of reference comprehension: For a given referring expression and scene, it predicts how easily the hearer will resolve the expression to the referent the speaker had in mind. The choice of a specific set of variables determines the aspects of the context that the model will take into account when making this prediction, and we can add further context variables or take some away. For example, the EqualCosts model we used as a baseline in the evaluation can be seen as an extreme variant of mSCRISP with no context variables. One might thus be able to assess the influence of individual variables on referential effectiveness by correlating the predictions that the model makes using different sets of variables with the comprehension behavior of human hearers.

Conversely, we might learn something about human reference production by examining human-produced referring expressions in the light of the ones that the model generates under the same setting. Human speakers do not always produce optimally effective references, and a question that arises is to what degree are human-produced references suboptimal. Because mSCRISP always generates correct and distinguishing references according to its model of what the hearer knows about, it explores levels of optimality that go beyond avoiding inappropriate use of privileged ground (e.g., Wardlow Lane and Ferreira (2008)); it seeks to optimize a reference according to a more fine-grained account of the hearer’s conceptual accessibility of referents and distractors in a scene (e.g., Fukumura et al. (2010); Arts et al. (2011)). By comparing such
different degrees of optimality against human production, empirical research might be able to study the question of just how effective human references are from a new angle.

**Conclusion**

In this work we presented mSCRISP, a computational model that generates referring expressions that are directly optimized for effectiveness in situated context. Computational models of reference generation often approximate effectiveness as humanlike-ness, but there has been recent empirical evidence that human-produced referring expressions may not always be optimally effective. Our model therefore learns to recognize human-produced referring expressions that are effective, and only aims at reproducing those. Because it recomputes the estimated contribution of each attribute of a referent to effectiveness based on the current situated context, mSCRISP does not rely on inflexible attribute preference orders like other state-of-the-art approaches. We have shown that mSCRISP indeed generates more effective referring expressions than baseline models, both in automatic and in full human task performance evaluations. The model formalizes the notion of situated context and could serve as a methodological framework for empirical research on referential effectiveness.
Figure 1

Figure 1: A simplified example of a CRISP lexicon and the derivation of the referring expression “the red button” describing $b_1$. 
Figure 2

\textbf{red}(u, x):
\begin{itemize}
\item Precond: canadjoin(N, u), referent(u, x), red(x), \ldots
\item Effect: \(\forall y. (\neg \text{red}(y) \rightarrow \neg \text{distractor}(u, y))\), \ldots
\end{itemize}

\textbf{left}(u, x):
\begin{itemize}
\item Precond: \(\forall y. (\neg (\text{distractor}(u, y) \land \text{left-of}(y, x)))\),
\text{canadjoin}(N, u), \text{referent}(u, x), \ldots
\item Effect: \(\forall y. (\text{left-of}(x, y) \rightarrow \neg \text{distractor}(u, y))\), \ldots
\end{itemize}

\textbf{the-button}(u, x):
\begin{itemize}
\item Precond: subst(NP, u), referent(u, x), button(x), \ldots
\item Effect: \(\forall y. (\neg \text{button}(y) \rightarrow \neg \text{distractor}(u, y)), \neg \text{subst}(NP, u), \ldots\)
\end{itemize}

Figure 2: Simplified CRISP planning operators for the lexicon of Figure 1, as in Garoufi and Koller (2010). Predicates subst express that a syntax node is open for substitution, referent connect syntax nodes to the semantic individuals to which they refer, and canadjoin indicate the possibility of a tree adjoining the given syntax node.
Figure 3

Figure 3: Map of a virtual world from the GIVE-2 corpus.
Table 1

<table>
<thead>
<tr>
<th>Attribute type</th>
<th>Frequency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Absolute (color; e.g. “red”, “blue”)</td>
<td>79.83</td>
</tr>
<tr>
<td>2. Taxonomic (object type; e.g. “button”, “box”)</td>
<td>59.80</td>
</tr>
<tr>
<td>3. Viewer-centered (e.g. “on the right”, “the left one”)</td>
<td>19.33</td>
</tr>
<tr>
<td>4. Micro-level landmark intrinsic (spatial relation with respect to a movable landmark; e.g. “by the chair”, “next to the couch”)</td>
<td>17.37</td>
</tr>
<tr>
<td>5. Macro-level landmark intrinsic (spatial relation with respect to an immovable landmark; e.g. “close to door”, “on other side”)</td>
<td>8.54</td>
</tr>
<tr>
<td>6. Distractor intrinsic (e.g. “across from yellow button”, “to the left of the blue button”)</td>
<td>7.00</td>
</tr>
<tr>
<td>7. History of interaction (e.g. “same”, “from before”)</td>
<td>5.60</td>
</tr>
<tr>
<td>8. Visual focus (e.g. “that”, “in your view”)</td>
<td>5.32</td>
</tr>
<tr>
<td>9. Elimination (e.g. “other”, “wrong”)</td>
<td>4.48</td>
</tr>
<tr>
<td>10. Relative (e.g. “first”, “middle”)</td>
<td>4.34</td>
</tr>
</tbody>
</table>

Table 1: Attribute type annotations and their relative frequency (i.e., proportion of annotated references that contain an attribute of the given type) in the English edition of the GIVE-2 corpus. In this work, we focus on the six most frequent types.
### Table 2

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OBJECT RELATIONS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. <em>RoomSameTypeDisNum</em></td>
<td>the number of distractors of the same type as the referent in the room</td>
<td>numeric</td>
</tr>
<tr>
<td>2. <em>MicroLandmarkInRoom</em></td>
<td>whether there are any micro-level (i.e., movable) landmarks in the referent’s room</td>
<td>{0, 1}</td>
</tr>
<tr>
<td>3. <em>MacroLandmarkNearby</em></td>
<td>whether there are any macro-level (i.e., immovable) landmarks near the referent</td>
<td>{0, 1}</td>
</tr>
<tr>
<td><strong>SPATIO-VISUAL</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. <em>Distance</em></td>
<td>the distance (in GIVE space units; including turning distance) between the IF and the referent</td>
<td>numeric</td>
</tr>
<tr>
<td>5. <em>Angle</em></td>
<td>the angle (in radians) between the center of the IF’s field of view and the referent</td>
<td>numeric</td>
</tr>
<tr>
<td><strong>REFERENT’S DISTINCTIVENESS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. <em>ColorUnique</em></td>
<td>whether the referent’s color is unique (i.e., not shared by other objects) in the world</td>
<td>{0, 1}</td>
</tr>
<tr>
<td>7. <em>LandmarkTypeUnique</em></td>
<td>whether a landmark of unique type in the world exists in the referent’s room</td>
<td>{0, 1}</td>
</tr>
<tr>
<td><strong>INTERACTION HISTORY</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. <em>Round</em></td>
<td>the number of times the referent has become a target to press throughout a session</td>
<td>numeric</td>
</tr>
<tr>
<td>9. <em>ReferenceAttempt</em></td>
<td>the number of times the referent has been referred to in the same round</td>
<td>numeric</td>
</tr>
<tr>
<td>10. <em>SeenDeltaTime</em></td>
<td>the time elapsed (in seconds) since the referent was last seen by the IF</td>
<td>numeric</td>
</tr>
</tbody>
</table>

Table 2: Context variables of referential scenes.
Table 3

<table>
<thead>
<tr>
<th>$j$</th>
<th>$a_j$</th>
<th>$v_j(s)$</th>
<th>$cost(a_j)$</th>
<th>$cost(\text{non}-a_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Absolute</td>
<td>0.03</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>2.</td>
<td>Taxonomic</td>
<td>1.02</td>
<td>1.02</td>
<td>0.00</td>
</tr>
<tr>
<td>3.</td>
<td>Viewer-centered</td>
<td>−29.40</td>
<td>0.00</td>
<td>29.40</td>
</tr>
<tr>
<td>4.</td>
<td>Micro-level landmark intrinsic</td>
<td>3.41</td>
<td>3.41</td>
<td>0.00</td>
</tr>
<tr>
<td>5.</td>
<td>Macro-level landmark intrinsic</td>
<td>11.84</td>
<td>11.84</td>
<td>0.00</td>
</tr>
<tr>
<td>6.</td>
<td>Distractor intrinsic</td>
<td>−23.00</td>
<td>0.00</td>
<td>23.00</td>
</tr>
</tbody>
</table>

Table 3: Example of weights $v_j(s)$ in a scene $s$ and corresponding cost assignments for each attribute type $a_j$. 
Effective reference in situated context

Figure 4

\texttt{red}(u, x):
   \textbf{Precond:} canadjoin(N, u), referent(u, x), . . .
   \textbf{Effect:} \neg\text{needtodecide}(\text{absolute}, u), . . .
   \textbf{Cost:} \text{cost}(\text{absolute})

\texttt{non-absolute}(u):
   \textbf{Precond:} needtodecide(\text{absolute}, u)
   \textbf{Effect:} \neg\text{needtodecide}(\text{absolute}, u)
   \textbf{Cost:} \text{cost}(\text{non-absolute})

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

<table>
<thead>
<tr>
<th>IG</th>
<th>Referring expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>the green button on the left</td>
</tr>
<tr>
<td>MaxEnt</td>
<td>the button to the left of the picture</td>
</tr>
<tr>
<td>EqualCosts</td>
<td>the left button, to the left of the right button</td>
</tr>
<tr>
<td>mSCRISP</td>
<td>the button to the left of the picture</td>
</tr>
</tbody>
</table>

Table 4: Referring expressions produced by a human instruction giver, our model mSCRISP and the two baselines MaxEnt and EqualCosts in the bottom-left room of Figure 3.
Table 5

<table>
<thead>
<tr>
<th>IG</th>
<th>Prob. of high succ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>0.467***</td>
</tr>
<tr>
<td>MaxEnt</td>
<td>0.984**</td>
</tr>
<tr>
<td>EqualCosts</td>
<td>0.649***</td>
</tr>
<tr>
<td>mSCRISP</td>
<td>0.957</td>
</tr>
</tbody>
</table>

Table 5: Average probabilities of high successfulness. Differences to mSCRISP are significant at **$p < 0.01$, ***$p < 0.001$ (paired t-tests).
Table 6

<table>
<thead>
<tr>
<th>IG</th>
<th>low succ.</th>
<th>high succ.</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt</td>
<td>0.320***</td>
<td>0.449*</td>
<td>0.371***</td>
</tr>
<tr>
<td>EqualCosts</td>
<td>0.512</td>
<td>0.475</td>
<td>0.497</td>
</tr>
<tr>
<td>mSCRISP</td>
<td>0.457</td>
<td>0.519</td>
<td>0.482</td>
</tr>
</tbody>
</table>

# references 78 51 129

Table 6: Average DICE coefficients across datasets. Differences to mSCRISP are significant at *$p < 0.05$, ***$p < 0.001$ (paired t-tests).
Figure 5

Figure 5: Example of a reference situated in the context of a GIVE evaluation scene, as generated by mSCRISP.
Table 7

<table>
<thead>
<tr>
<th>IG</th>
<th>Resolution success</th>
<th>Successfulness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all</td>
<td>non-rephrased</td>
</tr>
<tr>
<td>EqualCosts</td>
<td>86%***</td>
<td>86%</td>
</tr>
<tr>
<td>mSCRISP</td>
<td>95%</td>
<td>89%</td>
</tr>
</tbody>
</table>

Table 7: Average resolution success and successfulness results in the shared task. Differences to mSCRISP are significant at ***$p < 0.001$ (Pearson’s $\chi^2$ test for resolution success rates; unpaired two-sample t-tests for the rest).
Table 8

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>CL</th>
<th>L</th>
<th>T</th>
<th>EqualCosts</th>
<th>mSCRISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rate</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Average error rate results as in Striegnitz et al. (2011), putting mSCRISP to comparison against EqualCosts and the six other systems participating in the shared task. Two systems do not share the same letter if the difference between them is significant ($p < 0.05$; ANOVA and post-hoc Tukey tests).
Effective reference in situated context

Notes

1 Though this corpus is written, for the sake of simplicity we use the terms “speaker” and “hearer” for both spoken and written language settings.

2 See also an interesting computational treatment of Clark and Wilkes-Gibbs’ collaborative reference model by Heeman and Hirst (1995). This earlier computational model addresses reference generation using methods from automated planning, as we also do in this work.

3 The GIVE-2 corpus is freely available and viewable online at: http://www.give-challenge.org/research/page.php?id=give-2-corpus.

4 Further information on the GIVE Challenge as well as evaluation results are available at: http://www.give-challenge.org/research.
Acknowledgments

The research reported of here was partly supported by the Collaborative Research Center “Information Structure: The Linguistic Means of Structuring Utterances, Sentences and Texts” (SFB 632) at the University of Potsdam. Part of this work has been presented at the 2011 European Workshop on Natural Language Generation (Garoufi and Koller, 2011a,b); we are thankful to the workshop’s participants for their feedback. We also thank Ivan Titov for fruitful discussions and Albert Gatt as well as our reviewers for their very helpful comments.
References


