Generating Referential Descriptions Under Conditions of Uncertainty

Helmut Horacek
Universität des Saarlandes
F.R. 6.2 Informatik
Postfach 151150, D-66041 Saarbrücken, Germany
email: horacek@cs.uni-sb.de

Abstract

Algorithms for generating referring expressions typically assume that an object in a scenary can be identified through a set of commonly agreed properties. This is a strong assumption, since in reality properties of objects may be perceived differently among people, due to a number of factors including vagueness, knowledge discrepancies, and limited perception capabilities. Taking these discrepancies into account, we reinterpret concepts of algorithms generating referring expressions in view of uncertainties about the appearance of objects. Our model includes two complementary measures of likelihood in object identification, and adapted property selection and termination criteria. The approach is relevant for situations with potential perception problems and for scenarios with knowledge discrepancies between conversants.

1 Introduction

Generating referring expressions is a traditional, standard task in natural language generation. Over the past two decades, a number of algorithms have been proposed which differ among each other in terms of efficiency and coverage. To the best of our knowledge, all algorithms share the assumption that objects can be identified by a description consisting of attribute values ascribed to these objects. Moreover, the results are specified in a way that implicitly assumes complete agreement about these properties, provided they are known to the audience. We feel that this assumption may be too strong in reality so that, for instance, a dialog system in which the reference generation algorithm is embedded is unlikely to behave adequately when a misunderstanding occurs due to a perception mismatch.

In this paper, we address this problem by incorporating measures to deal with uncertainties into a standard algorithm that generates referring expressions. In order to represent uncertainties, we propose two complementary measures expressing the likelihood of object identification. We define computation schemes for combining descriptions with boolean combinations of attribute values, and we extend the incremental standard reference generation algorithm by adapting property selection and termination criteria.

This paper is organized as follows. First, we motivate our approach in more detail. Then we introduce our method for representing aspects of uncertainty. We follow by illustrating the propagation of uncertainty assessments for several attribute values, including boolean combinations, and we give examples of the effects. Then we describe extensions to the incremental algorithm, and we discuss their impact.

2 Motivation

In the scope of this paper, we adopt the terminology originally formulated in [Dale 1988] and later used by several others. A *referential description* [Donellan 1966] serves the purpose of letting the hearer or reader identify a particular object or set of objects in a given situation. The referring expression to be generated is required to be a *distinguishing description*, that is a description of the enitties being referred to, but not to any other object in the *context set.* A context set is defined as the set of the entities the addressee is currently assumed to be attending to – this is similar to the set of entities in the focus spaces of the discourse focus stack in Grosz' and Sidner's [1986] theory of discourse structure. Moreover, the *contrast set* (or the set of *potential distractors* [McDonald 1981]) is defined to entail all elements of the *context set* except the *intended referents*.

Generating referring expressions is pursued since the eighties [Appelt 1985, Kronfeld 1986, Appelt and Kronfeld 1987]. Subsequent years were characterized by a debate about computational efficiency versus minimality of the elements appearing in the resulting referring expression [Dale 1988, Reiter 1990, Reiter and Dale 1992]. In the mid-nineties, this debate seemed to be settled in favor of the incremental approach [Dale and Reiter 1995] – motivated by results of psychological experiments [Levelt 1989, Pechmann 1989], certain non-minimal expressions are tolerated in favor of adopting the fast strategy of incrementally selecting ambiguity-reducing attributes from a domain-dependent preference list. Recently, algorithms have been applied to the identification of sets of objects rather than individuals [Bateman 1999, Stone 2000, Krahmer, v. Erk, and Verweg 2001], and the repertoire of descriptions has been extended to boolean combinations of attributes, including negations [van Deemter 2002]. To avoid the generation of redundant descriptions that is typical for incremental approaches, Gardent [2002] and Horacek [2003] proposed exhaustive resp. best-first searches.

All these procedures more or less share the design of the underlying knowledge base. Objects are conceived in terms of sets of attributes, each with an atomic value as its filler. Some models distinguish specializations of these values according to a taxonomic hierarchy, so that the most accurate value can be replaced by one of its generalizations if there are reasons to assume this alternative is preferable – due to insufficient knowledge attributed to the audience, or to prevent unintended implications. A few approaches also deal with relations to other objects, whose representation differs from that of attributes only by the reference to the related object. Typically, a user model is assumed to guide the choice among available descriptors; the user model expresses taxonomic knowledge attributed to the user, indicating for a descriptor whether it is known to the user or not.

While a knowledge base developed and interpreted in this manner is adequate for generating referring expressions in most application-relevant settings, there may be circumstances in which uncertainties are prominent, so that the simple boolean attribution of properties to objects becomes problematic and may prove insufficient. Uncertainties may manifest themselves in at least the following three factors:

- Uncertainty about *knowledge*
  There may not be sufficient evidence to assume that the user is or is not acquainted with a specific term. In fact, most of today's user model components assign some probability to statements about a user's knowledge or capabilities, for example on the basis of inferences obtained through a belief network [Pearl 1988].
- Uncertainty about *perception capabilities*
  There is an increasing number of applications with natural language interaction where the objects of the discourse do not appear on the computer screen (e.g., ubiquitous tools guiding a user in environments such as airports and tourist attraction areas, e.g., [Wahlster 2004]). In such situations, perception and recognition of object properties is much harder to assess; for example, the visibility of some object or of one of its parts may not be derivable with complete certainty.
- Uncertainty about *conceptual agreement*
  While ascribing a value to an attribute is straightforward for certain categories of attributes, problems may occur, e.g., in connection with vagueness. This concept may be relevant for a number of commonly used properties such as size and shape, and even with colors, transitions between adjacent color tones may not be firmly categorized as one of the two candidates.

To illustrate these manifestations of uncertainty, let us consider a scenario with three similar dogs, one of which is a bassett, which is also the intended referent. In addition, the bassett is brownish and has a long tail. The other two dogs have shorter tails and their skin is also brown, but with some white resp. black portions. Furthermore, we assume that the audience has little knowledge about dog specifics, that is, it is not very likely that they may recognize the intended referent as a bassett. We also assume that the tails of the dogs cannot be observed easily by the audience under the given local circumstances.

Hence, the three attributes “category”, “color”, and “tail length” each fall into one of the categories of uncertainty introduced above: the categorization of the intended referent as a bassett is associated with uncertainty about knowledge, the limited visibility which may not enable the spectators to see the tails of the dogs in each moment constitutes an uncertainty about perception capabilities, and the similarity of the dogs' colors may yield uncertainty about conceptual agreement, that is, it is doubtfull whether the descriptor “brownish” is attributed only to the intended referent or also to some of the other dogs in the given situation.

Apparently, these uncertainties have consequences on building human-adequate referring expressions, especially in contexts where most of the descriptors available are associated with some kind of uncertainty. Intuitively, we would expect people to produce referring expressions with several of these descriptors, being redundant in case they are all recognized, but also hoping that the identification will succeed if the audience can identify only some part of the overall description in the given situation. Moreover, we would expect people only to use descriptors that have some reasonable chance of being understood.

Unfortunately, traditional generation algorithms do not enable us to model such a behavior, since none of the options available does justice to the uncertainty involved. If a descriptor is modeled as applying to all entities (e.g., for “brownish”), it will never be chosen since it yields no discrimination. A similar consequence is obtained when the capabilities of the audience are interpreted pessimistically. Finally, if a descriptor is assumed to be understood, it might be chosen without considering any of the other candidates associated with uncertainty. Thus, modeling in the existing algorithms forces us to make crisp decisions, with strong impacts on the result of the algorithm. Redundant expressions motivated by uncertainties about recognition cannot be generated under any modeling alternative.

There are only a few computational approaches which address the problem of uncertainty about the recognition of referring expressions. For example, [Edmonds 1994] and [Heeman and Hirst 1995] describe both plan-based methods, where a vague and partial description is produced initially, which is narrowed and ultimately confirmed in the subsequent discourse. However, the documented examples do only emphasize incomplete, but never incorrect interpretations. An approach that fits better to our intentions is the work by Goodman [1987], which emphasizes reference identification and associated failures in task-oriented dialogs [Goodman 1986]. This case study demonstrates various impacts of limitations and discrepancies of expertise on referential identification: subjects exhibit uncertainty in identification, which manifests itself in tentative actions and changes of mind, they misinterpret descriptions (e.g., 'outlet' interpreted as 'hole'), and they may find no appropriate referent at all. In the latter case, subject even undertake attempts to *repair* an otherwise uninterpretable description by relaxing descriptors. In the following, we interpret some of these findings for our model of uncertainty, including a model of a repair mechanism.

3 Representing Uncertainties

Basically, our model of uncertainty combines the three kinds of uncertainty described in the previous section. Each of them is expressed in terms of a probability, associated with a triple consisting of an object, an attribute applicable to that object, and the value ascribed to this pair. The following probabilities each express the likelihood that the user recognizes a description correctly from the perspectives of:

p_K The user is acquainted with the terms mentioned

p_P The user can perceive the properties uttered

p_A The user agrees to the applicability of the terms used

In order to identify an intended referent successfully, all three factors must be assessed positively, so that the *probability of recognition p* becomes the product of these three probabilities. Since the individual properties refer to factors outside the scope of proper generation, we only deal with *p* in the scope of this paper, although it is clear that this assessment requires contributions from several sources.

The concept of using individual probabilities to represent manifestations of uncertainty is not only simple, it also fits to knowledge sources where data about these probabilities could be found. For example, user models, the potential sources for assessing p_K, typically assign assessments to user capabilities on the basis of belief networks. Similar considerations hold for representations of vague properties, which fall under the concept of term agreement. These properties can be modeled by fuzzy logic systems [Zadeh 1984, 1996], which allow for an interpretation in terms of a single probability value, representing the likelihood that a precise value is perceived as a given vague term.

The association of a probability with the applicability of a descriptor to an object not only expresses the somehow direct likelihood of success of this task, but the application of this likelihood to several candidate objects also gives an indication of the likelihood of success of the overall identification goal. If a descriptor is assumed to be associated with several candidate objects by the audience, with certain degrees typically different among these objects, several cases can be distinguished: (1) *correct identification*, where the audience relates the description only to those objects to which this descriptor indeed applies, (2) *misinterpretation*, where none of these objects, but others are associated with the descriptor by the audience, (3) *ambiguity*, which is a combination of (1) and (2), and, finally, (4) the case of an *uninterpretable* description, where the audience does not relate the descriptor to any of the candidate objects. In the last case, people are known to make an attempt to *repair* their unsuccessful interpretation, since they assume that the expression communicated is indeed intended to refer to some object or objects in the domain of discourse, according to the work by Goodman. In order to simulate the effect of this behavior, we compute the probability of the occurrence of an *uninterpretable* description, which we call the *repair factor*, and we increase the probability of identification of the candidate objects (which we call our *repair mechanism*), based on the amount of the repair factor and the context of these objects in the overall identification task.

$$R(k, p_1, \ldots, p_n) = \sum_{j=0}^{k-1} \sum_{m=j}^{\binom{n}{m}} \prod_{i=1}^{n} \left(F(f(j,m,n), p_i) \,|\, f(j,m,n) \,|\right)$$

f(j,m,n) recursively enumerates all combinations of *m* out of *n* elements (here: natural numbers 1 … *n*) and returns the *j*-th combination as a set of numbers $M=\{i_1,\ldots,i_m\}$ with $1 \le i_k \le n$ ($k= 1,\ldots,m$)

$$F(M,p_i) = \begin{cases} p_i & \text{if } i \in M \\ (1-p_i) & \text{if } i \notin M \end{cases}$$

Figure 1. Repair factor for insufficient recognition of k objects

In concrete terms, if we have *n* objects for which a descriptor is recognized with probability p_i for object *i*, the probability that none of the objects is recognized by a referring expression built from that descriptor is $\Pi(1-p_i)$, $(1 \le i \le n)$. Although this number tends to be small if there are several objects to which the description matches with some reasonable degree of confidence, the associated need for invoking a repair mechanism becomes increasingly urgent when further descriptors are added to the description built so far, as well as when the task is to identify multiple referents rather than a single one. For the case of 2 objects, the need for invoking a repair mechanism can be quantified by the *repair factor* $2\Pi_{i=1,n}(1-p_i)+\Sigma_{i=1,n}(p_i\Pi_{j=1,n(j\neq i)}(1-p_j))$. The general case, if needed, gets increasingly complex, as illustrated in Figure 1, for *k* objects to be identified, out of *n* candidates ($k \le n$).

Thus, for the likelihood of recognition failure, a mechanism is required that simulates identification repair under these conditions. Apart from the likelihood of failure, repair should be guided by potential confusability of objects in view of some given descriptor. Hence, while we think it is virtually impossible to confuse an animal and a piece of equipment, at least under any reasonable conditions of visibility, we assume that objects of some degree of appearance similarity (size and shape) may potentially be confused with each other. Hence, we consider a potentially confusable object a candidate for being interpreted as an intended referent in case a repair of a reference failure is required. Confusion in this sense may be interpreted in two ways: from the perspective of the speaker, those objects are candidates which the speaker thinks the hearer could confuse. From the hearers perspective, those objects are candidates which the hearer thinks the speaker might have confused in producing a badly interpretable description. Since the latter constellation corresponds to the situation present for repair attempts, we model potential candidates quasi “objectively” by incorporating annotations in the knowledge base. The dependency of user capabilities as assessed by a user model influences these assessments indirectly through the probability of recognition attributed to the user for each descriptor-object pair.

Determine probability of *identification* $(D, k, O_1, \ldots, O_n)$

$O_1, \ldots, O_m$ Objects to which descriptor D applies

$O_{m+1}, \ldots, O_n$ Objects to which repair with D is applicable

$p_i, \ldots, p_m$ Probability that D is recognized for O_i

Objects ordered along degrees of recognition confidence:

$$\forall i,j(1 \leq i, j \leq m): (p_i > p_j) \rightarrow (i > j)$$

$R_{prop} \leftarrow R(k, p_1, \ldots, p_m),\ i \leftarrow 1$

1. if $(i \leq m)$
   then $R_c \leftarrow \mathrm{Min}(R_{prop}/n, 1\text{-}p_i),\ p\text{-}id_i \leftarrow p_i + R_c$
   else $R_c \leftarrow R_{prop}/n,\ p\text{-}id_i \leftarrow R_c$
   endif
   if $(i<n)$
   then $i \leftarrow i+1,\ R_{prop} \leftarrow R_{prop} - R_c$, goto 1
   endif

Figure 2. Assessing identification probabilities including repair

In order to keep the repair mechanism simple, we approximate confusability of an object by augmenting its representation with annotations of all property-value combinations that do not apply to it, but which could somehow be perceived as holding for this object. The potentially large amount of data created this way can be significantly reduced by making use of inheritance. For example, one can state that blue and purple (physical) objects can be confused, by making annotations about confusability with blue for purple objects, and vice-versa. This annotation is then inherited to all entities that are specializations of (physical) objects.

The proper repair is then simulated by collecting all candidates to which the descriptor in question could arguably apply, and by assigning these candidates a probability of *identification through repair*, according to the repair factor, as assessed above. There are two kinds of candidates: (1) those to which the descriptor is recognized with some probability, and (2) those to which it could apply with some relaxation, that is, which contains a suitable confusability annotation. The repair factor, which is computed according to the schema in Figure 1, is then evenly distributed among these two sets of candidates, provided the added probabilities of recognition and repair do not get greater than 1 for some object; this can only be the case if the number of objects to identify is close to the number of candidates. In such a case, the extra amount is distributed recursively among the remaining candidates, always respecting the upper limit of 1. If the number of objects to identify even exceeds the number of candidates, the effect of the repair mechanism results in a modification of the number of objects to identify, reducing it to the number of available candidates. The computation of the probability of *identification through repair* is illustrated in Figure 2. Three examples in Figure 3 illustrate the effect of the repair mechanism in quantitative terms. They emphasize the relation between expectations about the number of objects to be identified and probabilities of identification.

For k objects to be identified out of n, judging identification by descriptor D, which may involve repair measures (D applies to m out of these n with probabilities $p_i, \ldots, p_m$)

1. k=1, n=4, m=2 (p_1 =0.8, p_2 =0.4): R_{prop} = 0.12
   $p\text{-}id_1 = 0.83$, $p\text{-}id_2 = 0.43$, $p\text{-}id_3 = p\text{-}id_4 = 0.03$
2. k=2, n=4, m=2 (p_1 =0.8, p_2 =0.4): R_{prop} = 0.96
   $p\text{-}id_1 = 1$, $p\text{-}id_2 = 0.6533$, $p\text{-}id_3 = p\text{-}id_4 = 0.2533$
3. k=3, n=4, m=2 (p_1 =0.8, p_2 =0.4): R_{prop} = 2.1
   $p\text{-}id_1 = 1$, $p\text{-}id_2 = 1$, $p\text{-}id_3 = p\text{-}id_4 = 0.65$

Figure 3. Examples of assessing identification probabilities

Specifically, the increasing contributions of the repair facility are shown, which will be even more pronounced with several attributes associated with limited recognition expectations. We will see this effect in context with building descriptor combination in the next section, as well as in the detailed exposition of an example in Appendix II.

4 Identifiability of Descriptor Compositions

Since a single descriptor is rarely sufficient for identifying one or several objects in scenarios of interesting complexity, boolean compositions of descriptors are generated for this purpose, conjunctions being required for building identifying expressions for single objects. Their probability of recognition is a simple extension of the case of single descriptors. If p_i is the probability of recognition of descriptor D_i for some object O, an expression consisting of several D_i (i=1,pn) is identified with O through recognition if all D_i are attributed to O. The probability of this coincidence amounts to the product of all probabilities Πp_i (i=1,pn).

The probability of identification through repair is computed by distributing the repair factor $R(k,P_1,\ldots,P_m)$, where each $P_j = \Pi p_{ji}$ (j=1,m;i=1,pn), among all objects qualifying for the repair measure. While this distribution is an equal one for the case of a single descriptor, apart from using the upper limit of 1 for the total probability, such an even distribution would not do full justice here. We propose to distribute the likelihood proportionally to the probabilities of recognition for each descriptor, which makes repair more likely applicable to those objects which are also more likely to be identified anyway. In order to perform this operation properly, “average” probabilities (*ap*) for only reparable descriptors must be estimated. Moreover, we want to favor repairs for objects which require fewer “average” probabilities for this computation, by incorporating a "scale-down factor” (*sdf*) for each additional repair. The computation schema is given in Figure 4. For concrete computations, we choose 0.5 for both factors *ap* and *sdf* – see the examples in Figure 5. The first one demonstrates the partitioning of the repair factor according to the number of attributes which require repair. Specifically, the first three objects get the same share of the repair factor, while the fourth object gets only half of it, since its identification is the only one which requires

Compute identification probability $(D_1,\ldots,D_{np},k,O_1,\ldots,O_n)$

$O_1,\ldots,O_m$ Objects to which all $D_{1,\ldots,np}$ are applicable
$O_{m+1},\ldots,O_n$ Objects with repair possible for all $D_{1,\ldots,np}$
$p_{i1},\ldots,p_{inp}$ Probability that $D_{1,\ldots,np}$ is attributed to O_i
Objects ordered along degrees of identification confidence:

$$\forall i,j(1\leq i,j\leq m): (\Pi_{l=1,np}p_{il} > \Pi_{l=1,np}p_{jl}) \longrightarrow (i > j)$$

```
for i from 1 to n do
    P_i ← 1, sdf_i ← 1/sdf
    for j from 1 to np do
        if p_ij > 0
            then P_i ← P_i p_ij
            else P_i ← P_i ap, sdf_i ← sdf_i sdf
        endif
    endfor
endfor
R_prop ← R(k,P_i,…,P_m), i ← 1, P ← Σ_{i=1,n} P_i
1. if (i≤m)
    then R_c ← Min(R_prop(P_i/P),1-P_i), p-id_i ← P_i+R_c
    else R_c ← R_prop(P_i sdf_i/P), p-id_i ← R_c
endif
if (i<n)
    then i ← i+1, R_prop ← R_prop - R_c, goto 1
endif
```

Figure 4. Identification probabilities for several descriptors

For k objects to be identified out of n, judging identification by np descriptors D, at least repair possible for all (D_j applies to object i with probability p_{ji}, $\forall$i≤m: $p_{ji} > 0$)

1. k=1, n=4, m=1, np=2 (p_{11} =0.5, p_{21} =0.5, p_{12} =0.5, p_{22} =0, p_{13} =0, p_{23} =0.5, p_{14} =0, p_{24} =0): R_{prop} = 0.75
 $p\text{-}id_1$ = 0.464, $p\text{-}id_2$ = 0.214, $p\text{-}id_3$ = 0.214, $p\text{-}id_4$ = 0.107
2. k=2, n=3, m=1, np=2 (p_{11} =0.5, p_{21} =0.6, p_{12} =0.6, p_{22} =0.5, p_{13} =0, p_{23} =0.55): R_{prop} = 1.4
 $p\text{-}id_1$ = $p\text{-}id_2$ = 0.766, $p\text{-}id_3$ = 0.466
3. k=1, n=2, m=1, np=3 (p_{11} =0.5, p_{21} =0.5, p_{31} =0.5, p_{12} =0.9, p_{22} =0.9, p_{32} =0): R_{prop} = 0.875
 $p\text{-}id_1$ = 0.331, $p\text{-}id_2$ = 0.668

Figure 5. Examples of assessing identification probabilities

repair regarding two descriptors. The second example features the impact of multiple intended referents on the repair factor, which increases the probabilities of identification substantially. The last example illustrates the compensative effect between comparably low probabilities of recognition and higher ones in connection with the requirement of using the repair facility. Specifically, this example demonstrates that the probability of identification for an object (the second one) that is only identifiable through the repair mechanism can even become higher than the probability of identification for an object (the second one) that does not require repair for being identified. However, such an effect is only possible in the context of descriptors applicable with some degree of confidence to both candidates, but strongly favoring the object whose identification relies on the repair mechanism due to mismatch with another descriptor. This is the most critical effect in choosing descriptors.

The incorproation of disjunctions and negations is more local, since this extension only generalizes the probability of recognition of a single property. This is because these operators appear only in embedded boolean combinations [van Deemter 2002], which are the basis for building larger varieties of expressions [Horacek 2004]. For disjunctions of two descriptors with associated probabilities p_1 and p_2, the joint probability amounts to $p_1+p_2-p_1p_2$, assuming independence, which is quite normal for descriptors originating from distinct properties. For some properties, prominently those associated with vagueness, building disjunctions of descriptors originating from the same property may be beneficial. For example, disjunctions of similar colors or shapes may reduce the uncertainty through combining the identifiability of both. A simple way to model this constellation is by assigning probabilities to the set of applicable values so that their sum does not exceed 1, thereby modeling exclusion of the co-occurrence of more than one value. Consequently, the associated probabilities can simply be added. Propagation of the "confusable" annotation is treated similarly – if at least one of the descriptors is marked as "confusable", this also holds for the disjunction. For dealing with negation, the probability of identification is simply inverted (1-p). The treatment of the "confusable" annotation, however, is a bit problematic. The invertion operation needs modification through anticipating the amount of the repair factor, but this cannot be done locally. Therefore, this factor, *rf*, must be estimated in advance. For concrete computations we use a value of 0.1, so that $\neg p$ for a "confusable" p amounts to 0.9.

5 An Algorithm Incorporating Uncertainties

In this section, we describe extensions to the algorithm by Dale and Reiter [1995] that take into account the measures addressing uncertainty introduced in previous sections. This reference algorithm takes an intended referent r (the generalization to several referents is straightforward), the attributes P that describe r, and a contrast set C, and incrementally builds an identifying description L, if possible. The algorithm assumes an environment with three interface functions: *BasicLevelValue*, accessing basic level categories of objects [Rosch 1978], *MoreSpecificValue* for accessing incrementally specialized values of an attribute according to a taxonomic hierarchy, and *UserKnows* for judging whether the user is familiar with the attribute value of an object.

The algorithm basically iterates over the attributes P, according to some predetermined ordering which reflects preferences in the domain of application. For each attribute in P, a value assumed to be known to the user is determined, so that this value describes the intended referent and rules out at least one potential distractor which is still in the contrast set C in the iteration step considered. If such a value can be found, a pair consisting of the attribute and this value is included in the identifying description L. This step is repeated until the list P is exhausted or a distinguishing description is found, that is, the contrast set C is empty. Unless the distinguishing description L does not contain a descriptor expressible as a head noun, such a descriptor is added. Choosing the value of an attribute is done by an embedded iteration. It starts with the basic level value attributed to r, after which more specific values also attributed to r and assumed to be known to the user are tested for their discriminatory power. Finally, the least specific value that excludes the largest number of potential distractors and is known to the user is chosen. The schema of this procedure is given in Appendix I. The only modification we have done to the original version is the result of L as a non-distinguishing description in case of identification failure.

The algorithm by Dale and Reiter contains the principal operations that also other algorithms for generating referring expressions apply. The extension to boolean combinations of descriptors by van Deemter is essentially realized as an iteration around the Dale and Reiter algorithm, through building increasingly complex combinations, which other control regimes generate and maintain more effectively.

In order to control effects of facilities dealing with uncertainty, the extended algorithm has four control parameters:

- p_{min}, the minimal *probability of recognition* required for an attribute-value pair applicable to the intended referent, to justify its inclusion in the description,
- Δp_1, the minimal *improvement* in terms of probability of identification of the intended referent over a potential distractor obtained through an additional attribute-value pair,
- Δp_2, the minimal *preference* in terms of probability of identification of the intended referent over all potential distractors obtained through a description, and
- *Complexity-limit,* an upper bound on the *number of descriptors* collected in the distinguishing description.

In order to incorporate our concepts of representing uncertainty in this algorithm, we have to replace the interface functions which access crisp data and we must modify yes-no decisions. These enhancements concern:

- the decision about whether a descriptor excludes a potential distractor (in the function *RulesOut*),
- the choice of a value for an attribute (in the function *FindBestvalue*), and
- the termination of the overall procedure (in the function *MakeReferringExpression*)

Modifications of the reference algorithm are given in detail in the extended version in Appendix I – some lines are marked by labels [Ni] for references from the text. Expressions of the form pr(r,L) compute the probability of identification of referent r through the description L, according to the schema described in the previous sections.

Under conditions of uncertainty, determining whether a descriptor excludes a potential distractor may become a proper decision rather than a mere computation. A clear-cut case is only present if the repair facility is not applicable to one of the members of the contrast set, so that its associated probability of identification amounts to 0. This condition replaces the criterion that the user must know that this descriptor does not apply to some potential distractor in the function *RulesOut* [N7]. However, it would be a rather restrictive strategy to accept only those descriptors which definitely exclude a potential distractor. In fact, none of the descriptors that make up the example in Appendix II yield such a crisp discrimination. In addition to that, a descriptor is also valuable if it contributes to a better identification of the intended referent by increasing the difference to a potential distractor in the associated probabilities of identification by a significant margin (Δp_1). This criterion is added to the crisp criterion described above, encapsulated in the function *Dominate* [N8], which is used for this decision instead of the function *RulesOut* [N2]. The idea is that subsequently chosen descriptors have comparable effects on the identification of some of the other potential distractors, so that the intended referent ultimately gains over all of them. The significance of this margin must be tuned in such a way that the gain over some potential distractors is not outweighted by a loss over some other potential distractors.

The suitability of a value for an attribute depends on two factors associated with uncertainty: the probability of recognition associated with that value for the present user, and the effect of this value on excluding elements from the set of potential distractors. These two factors have adverse effects: while a more specific value has the potential of excluding an increasing number of potential distractors, its probability of recognition when applied to the intended referent may be lower than that of a less specific value. Consequently, it is not necessarily the case that an improved discriminatory power leads to a better overall effect. Hence, the choice of a value requires a minimal probability of recognition (p_{min}, [N6]), and calls to *Dominate* replace calls to *RulesOut*. Additional variants of descriptors can be generated by enhancing the interface function *MoreSpecificValue,* also building disjunctions of values excluding each other, to cover cases described at the end of Section 4, that is, building disjunctions of descriptors by composing descriptors (possibly vague ones) that cover adjacent value ranges.

The third factor, the termination criterion, is adapted to uncertainties by enhancing it in two ways: (1) a *complexity limit* is applied to the specifications in the description L [N3]; while this cut-off may serve practical considerations also without conditions of uncertainty (for a partitioning into sequences of descriptions [Horacek 2004]), it gains on relevance in uncertain environments. (2) a certain degree of being *Dominant* in the probability of identification over all potential distractors is considered sufficient (Δp_2, [N4]) rather than requiring the ultimate exclusion of all potential distractors. Finally, the conditions under which descriptors are selected, give rise to an optional optimization step. The prerequisite for this step is the distinction between descriptors which definitively exclude at least one potential distractor (L_{ro} in the extended algorithm, [N1]) and others which only affect their associated probabilities of identification, but do not make them 0. Then all subsets of the description built which contain at least L_{ro} are examined [N5] whether they yield a better preference over all potential distractors in terms of their probabilities of identification [N9]. Through this measure, an early chosen descriptor with a probability of identification lower for the intended referent than for some potential distractors can finally be discarded, provided the discriminating effect on other potential distractors is also achieved by later chosen descriptors. In the example in Appendix II, all descriptors are categorized as optional ones, but for the one expressing the head noun – which is precisely the reason why it is not optional.

Altogether, the algorithm selects descriptors which either exclude some potential distractors definitively, makes some of them rely on the repair mechanism, or simply increases the probability of identification of the intended referent considerably in comparison to elements of the contrast set. While this selection process works reasonably in most cases, it may turn out as problematic when several of the descriptors chosen are associated with limited probabilities of recognition for the intended referent in comparison to potential distractors not completely excluded. As a consequence, these potential distractors may be judged superior in terms of the probability of identification even though they rely on the repair mechanism (see example 3 in Figure 5). This risk can be circumvented by using a relatively high p_{min} parameter, but this measure may easily lead to the exclusion of an otherwise beneficial descriptor under normal conditions. An improvement can be obtained by the call to the procedure *Optimize*. If one of the first two descriptors used in example 3 in Figure 5 does not definitively exclude a potential distractor, the procedure *Optimize* tests descriptor combinations without it, and one of those may yield a better result – see also the example in Appendix II. A possible variations would be to allow just a single violation of the p_{min} restriction, for a descriptor with very good discriminatory power.

So far, we have only elaborated changes for incorporating uncertainty concepts to the reference algorithm per se. Handling boolean combinations of descriptors through applying the reference algorithm to increasingly complex combinations also works with uncertainties, since all computations required are defined. More difficulties arise with ambitious control regimes, which rely on cut-off techniques, in addition to the complexity cut-off, such as dominance and value cut-offs, as introduced in [Horacek 2004]. A complexity cut-off is already included in the extended reference algorithm. The two other cut-offs can be generalized, but this is likely to be associated with a significant loss of efficiency. In order for a descriptor to dominate another one, the dominating one must not only exclude all potential distractors that its competitor does, but it must also favor the intended referents over all potential distractors in terms of the associated probabilities of identification – this requirement reduces the application frequency of this cut-off considerably. A value cut-off, in turn, is applicable to a partial solution if a solution has already been found, and there are no descriptor combinations untested for the partial solution which may yield a solution with less complex specifications. This condition can also be met in the environment associated with uncertainties. In this environment, however, there is another factor that has an impact on the quality of the solution, that is the probability of identification, which cannot be assessed prior to actually choosing a descriptor and testing its effects.

6 Conclusion

In this paper, we have presented an approach for generating referential descriptions under conditions of uncertainty. The approach combines a proper recognition of objects associated with some degree of uncertainty, as well as identification through a repair mechanism, motivated by the need to identify objects even for descriptions that originally appear uninterpretable. On these lines, we have reinterpreted concepts of algorithms generating referring expressions in view of uncertainties about the appearance of objects. Incorporating measures of uncertainty in such an algorithm attacks strong assumptions and effects underlying most of the existing algorithms:

- They typically require crisp specifications concerning attribution of descriptors to referents and knowledge of the audience. Especially the connection to modern user models may require coarse-grained interpretations here.
- A single result is produced even if several reasonable variants exist, and this choice is implicitly determined by the preference ordering imposed on the descriptors.
- The interaction with other components of an NL generation system and an embedding dialog system is rather limited. Reference generation is typically conceived as a pure functional service, with no feedback, taking into account syntactic constraints, at best (e.g., [Horacek 1997]). An embedding dialog system has no chance to find out possible sources for an identification failure.

The algorithm incorporating measures to deal with uncertainties provides facilities to improve this situation:

- Specifications concerning attribution of descriptors to referents and knowledge of the audience can be done in a direct fashion, requiring no interpretations.
- There are some parameters to control the choice of descriptors, the conciseness and expected effectiveness of the result, including an afterwards optimization which only requires re-calculation of probabilities.
- The probabilities of identification associated with the intended referents and those potential distractors that fall under the repair facility give an indication about the likelihood of success of the identification task and also about potential sources for a failure. Moreover, the situation about probabilities and descriptors may suggest variants in building surface expressions, such as putting emphasis on a critical descriptor.

References

[Appelt 1985] Doug Appelt. Planning English Referring Expressions. *Artificial Intelligence* **26**:1-33, 1985.

[Appelt and Kronfeld 1987] Doug Appelt and Amichai Kronfeld. A Computational Model of Referring. In Proc. of the *10th International Joint Conference on Artificial Intelligence (IJCAI-87)*, pp. 640-647, Milano, Italy, 1987.

[Bateman 1999] John Bateman. Using Aggregation for Selecting Content when Generating Referring Expressions. In Proc. of the *37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pp. 127-134, University of Maryland, 1999.

[Dale 1988] Robert Dale. Generating Referring Expressions in a Domain of Objects and Processes. PhD Thesis, Centre for Cognitive Science, University of Edinburgh, 1988.

[Dale and Reiter 1995] Robert Dale and Ehud Reiter. Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science* **18**:233-263, 1995.

[Donellan 1966] K. Donellan. Reference and Definite Description. *Philosophical Review* **75**:281-304, 1966.

[Edmonds 1994] Phil Edmonds. Collaboration on Reference to Objects that are not Mutually Known. In Proc. of the *15th International Conference on Computational Linguistics* (*COLING-94)*, pp. 1118-1122, 1994.

[Gardent 2002] Claire Gardent. Generating Minimal Definite Descriptions. In Proc. of the *40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pp. 96-103, Philadelphia, Pennsylvania, 2002.

[Goodman 1986] Bradley Goodman. Reference Identification and Reference Identification Failures. *Computational Linguistics* **12**:273-305, 1986.

[Goodman 1987] Bradley Goodman. *Communication and Miscommunication*. Association of Computational Linguistics Series of Cambridge University Press, London, England, 1987.

[Grosz and Sidner 1986] Barbara Grosz and Candace Sidner. Attention, Intention, and the Structure of Discourse. *Computational Linguistics* **12**:175-206, 1986.

[Heeman and Hirst 1995] Peter Heeman and Graeme Hirst. Collaborating on Referring Expressions. *Computational Linguistics* **21**:351-382, 1995.

[Horacek 1997] Helmut Horacek. An Algorithm for Generating Referential Descriptions with Flexible Interfaces. In Proc. of the *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL'97)*, pp. 206-213, Madrid, Spain, 1997.

[Horacek 2003] Helmut Horacek. A Best-First Search Algorithm for Generating Referring Expressions. In Proc. of the *10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2003)*, Conference Companion (short paper), pp. 103-106, Budapest, Hungary, 2003.

[Horacek 2004] Helmut Horacek. On Referring to Sets of Objects Naturally. In Proc. of the *Third International Conference on Natural Language Generation (INLG-2004)*, pp. 70-79, Brockenhurst, UK, 2004.

[Krahmer, v. Erk and Verleg 2001] Emiel Krahmer, S. v. Erk, André Verleg. A Meta-Algorithm for the Generation of Referring Expressions. In Proc. of the 8th European Workshop on Netural Language Generation (*EWNLG-2001)*, pp. 29-39, Toulouse, France, 2001.

[Kronfeld 1986] Amichai Kronfeld. Donellan's Distinction and a Computational Model of Reference. In Proc. of the *24th Annual Meeting of the Association for Computational Linguistics (ACL-86)*, pp. 186-191, New York, NY, 1986.

[Levelt 1989] William Levelt. *Speaking: From Intention to Articulation*. MIT Press, 1989.

[McDonald 1981] David McDonald. Natural Language Generation as a Process of Decision Making under Constraints. PhD thesis, MIT, 1981.

[Pearl 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inferences*. Morgan Kaufman, San Mateo, California, 1988.

[Pechmann 1989] Thomas Pechmann. Incremental Speech Production and Referential Overspecification. *Linguistics* **27**:89-110, 1989.

[Reiter 1990] Ehud Reiter. The Computational Complexity of Avoiding Conversational Implicatures. In Proc. of the *28th Annual Meeting of the Association for Computational Linguistics (ACL-90)*, pp. 97-104, Pittsburgh, Pennsylvania, 1990.

[Reiter and Dale 1992] Ehud Reiter and Robert Dale. Generating Definite NP Referring Expressions. In Proc. of the 14th International Conference on Computational Lingustics (*COLING-92)*, pp. 232-238, Nantes, France, 1992.

[Rosch 1978] Eleanor Rosch. Principles of Categorization. In E. Rosch and B. Llyod (eds.) *Cognition and Categorization*, pp. 27-48, Hillsdale, NJ: Lawrence Erlbaum, 1978.

[Stone 2000] Matthew Stone. On Identifying Sets. In Proc. of the *First International Conference on Natural Language Generation (INLG-2000)*, pp. 116-123, Mitzpe Ramon, Israel, 2000.

[van Deemter 2002] Kees van Deemter. Generating Referring Expressions: Boolean Extensions of the Incremental Algorithm. *Computational Linguistics*, **28**(1):37-52, 2002.

[Wahlster 2004] Wolfgang Wahlster. *REAL: REssourcen-Adaptive Lokalisation*. Project in SFB 378, Saarland University, 2004.

[Zadeh 1984] Lofti Zadeh. Making Computers Think like People. *IEEE Spektrum*, **8**:26-32, 1984.

[Zadeh 1996] Lofti Zadeh. Fuzzy Logic and the Calculi of Fuzzy Rules and Fuzzy Graphs. *International Journal of Multi-Valued Logic*, **8**:1-39, 1996.

Appendix I: Reference Algorithm ([Dale and Reiter 1995], left) and Extended Algorithm (right)

```
MakeReferringExpression (r,C,P)
L ← {}
for each member A_i of list P do
   V = FindBestValue(r,A_i,BasicLevelValue(r,A_i))
   if RulesOut(<A_i,V>) ≠ nil
      then L ← L ∪ {<A_i,V>}
           C ← C - RulesOut(<A_i,V>)
   endif
   if C = {} then
      if <type,X> ∈ L for some X
         then return L          (an identifying description)
         else return L ∪ {<type,BasicLevelValue(r,type)>}
      endif
   endif
return L                        (a non-identifying description)

FindBestValue (r,A,initial-value)
if UserKnows(r,<A,initial-value>) = true
   then value ← initial-value
   else value ← no-value
endif
if (spec-value ← MoreSpecificValue(r,A,value)) ≠ nil ∧
   (new-value ← FindBestValue(r,A,spec-value)) ≠ nil ∧
   (|RulesOut(<A,new-value>)| > |RulesOut(<A,value>)|)
   then value ← new-value
endif
return value

RulesOut (A,V)
if V = no-value
   then return nil
   else return {x: x ∈ C ∧ UserKnows(x,<A,V>) = false}
endif
```

```
MakeReferringExpression (r,C,P)
L ← {}, L_ro ← {}                                          [N1]
for each member A_i of list P do
   V = FindBestValue(r,A_i,BasicLevelValue(r,A_i))
   if Dominate(A_i,V) ≠ nil                                [N2]
      then L ← L ∪ {<A_i,V>}
           C ← C - RulesOut(<A_i,V>)
           if RulesOut(<A_i,V>) ≠ nil
              then L_ro ← L_ro ∪ {<A_i,V>}
   endif  endif
   if (C = {}) ∨ (|L| > Complexity-limit) ∨                [N3]
      (∀x ∈ C: (pr(r,L) - pr(x,L)) > Δp_2) then            [N4]
      L ← Optimize(L,L_ro)                  (optional)     [N5]
      if <type,X> ∈ L for some X
         then return L  (a likely identifying description)
         else return L ∪ {<type,BasicLevelValue(r,type)>}
   endif  endif
return L                    (an unlikely identifying description)

FindBestValue (r,A,initial-value)
if pr(r,{<A,initial-value>}) > p_min                       [N6]
   then value ← initial-value
   else value ← no-value
endif
if (spec-value ← MoreSpecificValue(r,A,value)) ≠ nil ∧
   (new-value ← FindBestValue(r,A,spec-value)) ≠ nil ∧
   (|Dominate(A,new-value)| > |Dominate(A,value)|)
   then value ← new-value
endif
return value

RulesOut (A,V)
if V = no-value then return nil
   else return {x: x ∈ C ∧ (pr(x,L ∪ {<A,V>}) = 0)}      [N7]
endif

Dominate (A,V)
if V = no-value then return nil
else return {x: x ∈ C ∧ ((pr(x,L ∪ {<A,V>}) = 0) ∨
         ((pr(r,L ∪ {<A,V>}) - pr(x,L ∪ {<A,V>})) -       [N8]
         (pr(r,L) - pr(x,L))) > Δp_1)}
endif

Optimize (L_1,L_2)
L_opt ← L_1, pp_opt ← Min_{∀x ∈ C}(pr(r,L_1) - pr(x,L_1))   [N9]
forall L (L_1 ⊇ L ⊇ L_2) do
   pp ← Min_{∀x ∈ C}(pr(r,L) - pr(x,L))
   if pp_opt < pp then L_opt ← L, pp_opt ← pp endif
endforall
return L_opt
```

Appendix II: An Example of the Extended Algorithm at Work

In this section, we demonstrate the functionality of the extended algorithm by an example that illustrates several features of this algorithm. As in the discussion in Section 2, the scenario consists of three similar dogs, one of which is a bassett, which is also the intended referent. In addition, the bassett is brownish and has a long tail. The other two dogs have shorter tails and their skin is also brownish, but with some white resp. black portions, which makes the descriptor 'brownish' less appropriate than for the bassett.

To make the example suitable for our purposes, the audience is assumed to have little knowledge about dog specifics, that is, they may recognize the intended referent as a bassett, but this is not very likely (we assume the category identification has a likelihood of 30%). In addition, it is assumed that the tails of the dogs, specifically the one of the bassett, cannot be observed easily by the audience (again, we assume the recognition has a likelihood of 30%). Both, dog category and tail length are potentially confusable for all dogs. These properties and associated probabilities of identification as listed below.

Objects	*Attributes* category	color	tail-length
dog_1	bassett	brownish	long
dog_2	dog	brown-white	short
dog_3	dog	brown-black	short

Probabilities of identification (per attribute-value and object)

bassett:	p(dog_1) = 0.3, p(dog_2) = 0.0, p(dog_3) = 0.0
brownish:	p(dog_1) = 0.9, p(dog_2) = 0.8, p(dog_3) = 0.8
long-tail:	p(dog_1) = 0.3, p(dog_2) = 0.0, p(dog_3) = 0.0

Hence, the intended referent r is {dog_1}, and the contrast set C is {dog_2, dog_3}. For demonstration purposes, we choose the following parameterizations:

- The attributes are considered according to the ordered preference list P = (color, category, tail-length), which in some sense reflects the ease of perception of color
- We choose 5% (0.05) for Δp_1, which indicates sufficient dominance, and 30% (0.3) for Δp_{min}, which indicates sufficient identification potenial (it always succeeds in the example); similarly, we choose 50% (0.5) for Δp_2 which indicates sufficient discrimination (it never succeeds here, hence all descriptors are tried)
- We allow a maximum complexity of 3 descriptors, so that this cut-off criterion does not apply in our simple example, and we use the optional optimization step
- In order to compute probabilities of identification, we need to choose a values for the ”scale-down factor”, which will be 0.5, as mentioned in Section 4. Since no disjunctions and negations of descriptors are needed for our example, no further parameters are required.

We now illustrate the generation process step by step.

In the first step, “brownish” is chosen as the value of the attribute “color” of dog_1, and its contribution to discriminate the intended referent from the elements of the contrast set is checked. To start with, its identification potential, 0.9, is far higher than p_{min} (0.25).Since this descriptor is also applicable to both other dogs, dog_2 and dog_3, but with lower probability, these two objects still remain in the contrast set. Despite this limited discrimination, the attribute is chosen because it achieves more than the minimal dominance required: 0.9 - 0.8 equals 0.1, which is higher than Δp_2 (0.05). Thus, the situation after step 1 is as follows (we neglect the repair factor, which is as low as 0.1 x 0.2 x 0.2):

pr(dog_1) = 0.9, pr(dog_2) = pr(dog_3) = 0.8
Δpr = pr(dog_1) - Max(pr(dog_2), pr(dog_3)) = 0.1 > Δp_1

In the next step, “bassett” is chosen as the value of the attribute “category” of dog_1, and again its contribution to discriminate the intended referent from the elements of the contrast set is checked. Its identification potential, 0.3, is sufficient, since it is higher than p_{min} (0.25). This descriptor is not applicable to the other dogs, dog_2 and dog_3. Nevertheless, they still remain in the contrast set since they potentially are subject to the repair mechanism. The probabilities of identification are computed according to the schema in Figure 4: the product of the probabilities of "bassett" and "brownish" associated with dog_1 yields 0.27. The repair factor, the complementing 0.73, is distributed evenly among all three dogs. Hence, the degree of dominance of this descriptor amounts to 0.27, which is higher than Δp_2 (0.05). Thus, the situation after step 2 is as follows:

pr(dog_1) = 0.513, pr(dog_2) = pr(dog_3) = 0.243
Δpr = pr(dog_1) - Max(pr(dog_2), pr(dog_3)) = 0.27 > Δp_1

In the last step, “long” is chosen as the value of the attribute “tail-length”. Again, its identification potential, 0.3, is sufficient, since it is higher than p_{min} (0.25). As in the previous step, dog_2 and dog_3 still remain in the contrast set since they potentially are subject to the repair mechanism. The product of the probabilities for dog_1 results from the previous one, multiplied by 0.3, which yields 0.081. The repair factor, the complementing 0.919, is distributed by giving two parts to dog_1 (the scale-down factor applies) and one part to each of the other dogs. Hence, the degree of dominance of this descriptor amounts to 0.31075, which is higher than Δp_2 (0.05), which gives the final situation:

pr(dog_1) = 0.5405, pr(dog_2) = pr(dog_3) = 0.22975
Δpr = pr(dog_1) - Max(pr(dog_2), pr(dog_3)) = 0.31075 > Δp_1

Optimization attempts show that “bassett” only is slightly inferior (pr(dog_1) = 0.53, pr(dog_2) = pr(dog_3) = 0.23, Δpr = 0.3), while “bassett” together with “long-tailed” is slightly superior (pr(dog_1) = 0.545, pr(dog_2) = pr(dog_3) = 0.225, Δpr = 0.32) to the combination of all descriptors. Hence, the example demonstrates benefits and risks are comparable when only limited discrimination is possible by each descriptor.