# A Fast Algorithm for the Generation of Referring Expressions

**Ehud Reiter**[*] **and Robert Dale**[†]
Department of Artificial Intelligence
University of Edinburgh
Edinburgh EH1 1HN
Scotland

## Abstract

We simplify previous work in the development of algorithms for the generation of referring expressions while at the same time taking account of psycholinguistic findings and transcript data. The result is a straightforward algorithm that is computationally tractable, sensitive to the preferences of human users, and reasonably domain-independent. We provide a specification of the resources a host system must provide in order to make use of the algorithm, and describe an implementation used in the IDAS system.

## Introduction

In previous work [Dal89,DH91,Rei90a,Rei90b] we have proposed algorithms for determining the content of referring expressions. Scrutiny of the psycholinguistics literature and transcripts of human dialogues shows that in a number of respects the behaviour of these algorithms does not correspond to what people do. In particular, as compared to these algorithms, human speakers pay far less attention to reducing the length of a referring expression, and far more attention to making sure they use attributes and values that human hearers can easily process; in the terms introduced in [Dal88,Dal89], hearers are more concerned with the principle of sensitivity than with the principle of efficiency. We have designed a new referring expression generation algorithm that is based on these observations, and believe that the new algorithm is more practical for real-world natural language generation systems than the algorithms we have previously proposed. In particular, the algorithm is:

- *fast:* its run-time is linear in the number of distractors, and independent of the number of possible modifiers;
- *sensitive to human preferences:* it attempts to use easily perceivable attributes and basic-level [Ros78] attribute values; and
- *domain-independent:* the core algorithm should work in any domain, once an appropriate knowledge base and user model has been set up.

A version of the algorithm has been implemented within the IDAS natural-language generation system [RML92], and it is performing satisfactorily.

The algorithm presented in this paper only generates definite noun phrases that identify an object that is in the current focus of attention. Algorithms and models that can be used to generate pronominal and *one*-anaphoric referring expressions have been presented elsewhere, e.g., [Sid81,GJW83,Dal89]. We have recently begun to look at the problem of generating referring expressions for objects that are not in the current focus of attention; this is discussed in the section on Future Work.

## Background

### Distinguishing Descriptions

The term 'referring expression' has been used by different people to mean different things. In this paper, we define a referring expression in intentional terms: a noun phrase is considered to be a referring expression if and only if its only communicative purpose is to identify an object to the hearer. In Kronfeld's terminology [Kro86], we only use the modal aspect of Donellan's distinction between attributive and referential descriptions [Don66]; we consider a noun phrase to be referential if it is *intended* to identify the object it describes to the hearer, and attributive if it is *intended* to communicate information about that object to the hearer. This usage is similar to that adopted by Reiter [Rei90b] and Dale and Haddock [DH91], but differs from the terminology used by Appelt [App85], who allowed 'referring expressions' to satisfy any communicative goal that could be stated in the underlying logical framework.

We here follow Dale and Haddock [DH91] in assuming that a referring expression satisfies the referential communicative goal if it is a **distinguishing description**, i.e., if it is an accurate description of the entity being referred to, but not of any other object in the current **context set**. We define the context set to be the set of entities that the hearer is currently assumed to be attending to; this is similar

[*] Supported by SERC grant GR/F/36750. E-mail address is E.Reiter@ed.ac.uk.

[†] Also of the Centre for Cognitive Science at the University of Edinburgh. E-mail address is R.Dale@ed.ac.uk.

to the notion of a discourse focus space [GS86]. We also define the **contrast set** to be all elements of the context set except the intended referent. The role of the components of a referring expression can then be regarded as 'ruling out' members of the contrast set. For example, if the speaker wished to identify a small black dog in a situation where the contrast set consisted of a large white dog and a small black cat, she might choose the adjective *black* in order to rule out the white dog and the head noun *dog* in order to rule out the cat; this results in the referring expression *the black dog*, which matches the intended referent but no other object in the current context. *The small dog* would also be a successful referring expression in this context, under the distinguishing description model.

## Unnecessary Modifiers

A referring expression must communicate enough information to be able to uniquely identify the intended referent in the current discourse context (i.e., it must adhere to the principle of adequacy [Dal88,Dal89]). But this is not the only constraint a good referring expression must obey; it is clear that many referring expressions that meet this constraint are inappropriate because they convey incorrect and unwanted **conversational implicatures** [Gri75,Rei90a] to a human hearer.

One source of such false implicatures can be the presence of redundant or otherwise unnecessary modifiers in a referring expression. For example, consider two possible referring expressions that a speaker might use to request that a hearer sit by a table:

(1) a. Sit by *the table.*
b. Sit by *the brown wooden table.*

If the context was such that only one table was visible, and this table was brown and made of wood, utterances (1a) and (1b) would both be distinguishing descriptions that unambiguously identified the intended referent to the hearer; a hearer who heard either utterance would know where he was supposed to sit. However, a hearer who heard utterance (1b) in such a context might make the additional inference that it was important to the discourse that the table was brown and made of wood; for, the hearer might reason, why else would the speaker include information about the table's colour and material that was not necessary for the reference task? This inference is an example of a conversational implicature caused by a violation of Grice's maxim of Quantity [Gri75].

## Inappropriate Modifiers

Unwanted conversational implicatures can also be caused by the use of overly specific or otherwise unexpected modifiers. One example is as follows:

(2) a. Look at *the dog.*
b. Look at *the pit bull.*

In a context where there is only one dog present, the hearer would normally expect utterance (2a) to be used, since dog is a **basic-level** class [Ros78] for most native speakers of English. Hence the use of utterance (2b) might implicate to the hearer that the speaker thought it was relevant that the animal was a pit bull and not some other kind of dog [Cru77], perhaps because the speaker wished to warn the hearer that the animal might be dangerous; if the speaker had no such intention, she should avoid using utterance (2b), despite the fact that it fulfills the referential communicative goal.

## Previous Work

In previous work [Dal89,DH91,Rei90a,Rei90b] we have noted that the presence of extra information in a referring expression can lead the hearer to make false implicatures, and therefore concluded that a referring-expression generation system should make a strong attempt to ensure that generated referring expressions do not include unnecessary information, either as superfluous NP modifiers or as overly-specific head nouns or attribute values. Dale [Dal88,Dal89,DH91] has suggested doing this by requiring the generation system to produce **minimal** distinguishing descriptions, i.e., distinguishing descriptions that include as few attributes of the intended referent as possible. Reiter [Rei90a,Rei90b] has pointed out that this task is in fact NP-Hard, and has proposed instead that referring expressions should obey three rules:

**No Unnecessary Components:** all components of a referring expression must be necessary to fulfill the referential goal. For example, *the small black dog* is not acceptable if *the black dog* is a distinguishing description, since this means *small* is an unnecessary component.

**Local Brevity:** it should not be possible to produce a shorter referring expression by replacing a set of existing modifiers by a single new modifier. For example, *the sleeping female dog* should not be used if *the small dog* is a distinguishing description, since the two modifiers *sleeping* and *female* can be replaced by the single modifier *small.*

**Lexical Preference:** this is an extension of the basic-level preference proposed by Cruse [Cru77]; more details are given in [Rei91].

A referring expression that meets Reiter's constraints can be found in polynomial time if the lexical preference relation meets certain conditions [Rei90a]; such a referring expression can not, however, always be found in linear time.

# Psychological and Transcript Data

## Psychological Evidence

Subsequent to performing the above research, we have looked in some detail at the psychological literature on human generation of referring expressions. This research (e.g., [FO75,Whi76,Son85, Pec89]; [Lev89, pages 129–134] is a useful summary of much of this work) clearly shows that in many cases human speakers do include unnecessary modifiers in referring expressions; this presumably implies that in many cases human hearers do not make implicatures from the presence of unnecessary modifiers. For example, if human subjects are shown a picture of a white bird, a black cup, and a white cup, and are asked to identify the white bird, they frequently say *the white bird*, even though just *the bird* would have been sufficient in this case.

A partial explanation for this use of redundancy may be that human speakers generate referring expressions incrementally [Pec89]. An incremental generation algorithm cannot always detect unnecessary modifiers; in the above example, for instance, one could imagine the algorithm choosing the adjective *white* to rule out the black cup, and then the noun *bird* in order to rule out the white cup, without then erasing *white* because the black cup is also ruled out by *bird*.

Another explanation of redundancy might involve the speaker's desire to make it easier for the hearer to identify the object; the speaker might believe, for example, that it is easier for the hearer to identify a *white bird* than a *bird*, since colour may be more immediately perceptible than shape.[1].

Both of the above explanations primarily justify adjectives that have some discriminatory power even if they are redundant in this particular context. In the above example, for instance, *white* possesses some discriminatory power since it rules out the black cup, even though it does happen to be redundant in the expression *the white bird*. It would be harder for either of the above factors to explain the use of a modifier with no discriminatory power, e.g., the use of *white* if all objects in the contrast set were white. There is some psychological research (e.g., [FO75]) that suggests that human speakers do not use modifiers that have no discriminatory power, but this research is probably not conclusive.

The argument can be made that psychological realism is not the most important constraint for generation algorithms; the goal of such algorithms should be to produce referring expressions that human hearers will understand, rather than referring expressions that human speakers would utter. The fact that human speakers include redundant modifiers in referring expressions does not mean that NL generation systems are also required to include such modifiers; there is nothing in principle wrong with building generation systems that perform more optimizations of their output than human speakers. On the other hand, if such beyond-human-speaker optimizations are computationally expensive and require complex algorithms, they may not be worth performing; they are clearly unnecessary in some sense, after all, since human speakers do not perform them.

## Transcript Analysis

In addition to the psychological literature review, we have also examined a transcript of a dialogue between two humans performing an assembly task.[2] We were particularly interested in questions of modifier choice; if a discriminating description can be formed by adding any one of several modifiers to a head noun, which modifier should be used? In particular,

1. Which attribute should be used? E.g., is it better to generate *the small dog*, *the black dog*, or *the female dog*, if these are discriminating descriptions but just *the dog* is not?
2. Is it preferable to add a modifier or to use a more specific head noun? E.g., is it better to say *the small dog* or *the chihuahua*?
3. Should relative or absolute adjectives be used? E.g., is it better to say *the small dog* or *the one foot high dog*?

In our analysis, we observed several phenomena which we believe may generalise to other situations involving spoken, face-to-face language:

1. Human speakers prefer to use adjectives that communicate size, shape, or colour in referring expressions. In the above examples, for instance, a human speaker would probably prefer *the black dog* and *the small dog* over *the female dog*.
2. Human hearers sometimes have trouble determining if an object belongs to a specialized class. In the above example, for instance, *the chihuahua* should only be used if the speaker is certain the hearer is capable of distinguishing chihuahuas from other types of dogs. If there is any doubt about the hearer's ability to do this, adding an explicit modifier (e.g., *the small dog*) is a better strategy than using a specialized head noun.
3. Human speakers seem to prefer to use relative adjectives, and human hearers seem to have less trouble understanding them. However, human-written instructional texts sometimes use absolute adjectives instead of relative ones; this may be a consequence of the fact that writers cannot predict the context their text will be read in, and hence how readers will interpret relative adjectives. In the above example, therefore, a speaker would be expected to use *the small dog*, but a writer might use *the one foot high dog*.

[1] Another possible explanation is that speakers may in some cases use precompiled 'reference scripts' instead of computing a referring expression from scratch; such reference scripts specify a set of attributes that are included as a group in a referring expression, even if some members of the group have no discriminatory power in the current context

[2] The transcript was made by Phil Agre and John Batali, from a videotape taken by Candy Sidner. We are very grateful to them for allowing us to use it.

## The Algorithm

Based on the above considerations, we have created a new algorithm for generating referring expressions. This algorithm is simpler and faster than the algorithms proposed in [Dal89,Rei90a] because it performs much less length-oriented optimization of its output; we now believe that the level of optimization suggested in [Dal89,Rei90a] was unnecessary and psycholinguistically implausible. The algorithm has been implemented as part of a larger natural-language generation system, and we are pleased with its performance to date.

### Assumptions about the Knowledge Base

Our algorithm is intended to be reasonably domain-independent. We do, however, make some assumptions about the structure of the host system's underlying knowledge base, and require that certain interface functions be provided.

In particular, we assume that:

- Every entity is characterised in terms of a collection of **attributes** and their **values**. An attribute–value pair is what is sometimes thought of as a property; an example is ⟨colour, red⟩.
- Every entity has as one of its attributes some **type**. This is a special attribute that corresponds to the kinds of properties that are typically realized by head nouns; an example is ⟨type, dog⟩.
- The knowledge base may organize some attribute values in a subsumption taxonomy (e.g., as is done in KL-ONE [BS85] and related KR systems). Such a taxonomy might record, for example, that animal subsumes dog, and that red subsumes scarlet. For such taxonomically-organized values, the knowledge-base or an associated user-model should specify which level of the taxonomy is basic-level for the current user.

We require that the following interface functions be provided:

value(object,attribute) returns the value (if any) that an attribute has for a particular object. Value should return the most specific possible value for this attribute, e.g., chihuahua instead of dog, and scarlet instead of red.

taxonomy-children(value) returns the immediate children of a value in the taxonomy. For example, taxonomy-children(animal) might be the set {dog, cat, horse, ...}.

basic-level-value(object,attribute) returns the basic-level value of an attribute of an object. For example, basic-level-value(Garfield, type) might be cat. The knowledge-representation system should in principle allow different basic-level classes to be specified for different users [Ros78,Rei91].

user-knows(object, attribute-value-pair) returns true if the user knows or can easily determine (e.g., by direct visual perception) that the attribute–value pair applies to the object; false if the user knows or can easily determine that the attribute–value pair does *not* apply to the object; and unknown otherwise. For example, if object $x$ had the attribute–value pair ⟨type, chihuahua⟩, and the user was capable of distinguishing dogs from cats, then user-knows($x$, ⟨type, dog⟩) would be true, while user-knows($x$, ⟨type, cat⟩) would be false. If the user was not, however, capable of distinguishing different breeds of dogs, and had no prior knowledge of $x$'s breed, then user-knows($x$, ⟨type, chihuahua⟩) and user-knows($x$, ⟨type, poodle⟩) would both return unknown, since the user would not know or be able to easily determine whether $x$ was a chihuahua, poodle, or some other breed of dog.

Finally, we assume that the global variable *preferred-attributes* lists the attributes that human speakers and hearers prefer (e.g., type, size, shape, and colour in the assembly task transcript mentioned above). These attributes should be listed in order of preference, with the most preferable attribute first. The elements of this list and their order will vary with the domain, and should be determined by empirical investigation.

### Inputs to the Algorithm

In order to construct a reference to a particular entity, the host system must provide:

- a symbol corresponding to the intended referent; and
- a list of symbols corresponding to the members of the contrast set (i.e., the other entities in focus, besides the intended referent).

The algorithm returns a list of attribute–value pairs that correspond to the semantic content of the referring expression to be realized. This list can then be converted into an SPL [Kas89] term, as is done in the IDAS implementation; it can also be converted into a **recoverable semantic structure** of the kind used in Dale's EPICURE system [Dal88,Dal89].

### The Algorithm

In general terms, the algorithm iterates through the attributes in *preferred-attributes*. For each attribute, it checks if specifying a value for it would rule out at least one member of the contrast set that has not already been ruled out; if so, this attribute is added to the referring set, with a value that is known to the user, rules out as many contrast set members as possible, and, subject to these constraints, is as close as possible to the basic-level value. The process of adding attribute–value pairs continues until a referring expression has been formed that rules out every member of the contrast set. There is no backtracking; once an attribute–value pair has been added to the referring expression, it is not removed even if the addition of subsequent attribute–value pairs make it unnecessary. A head noun (i.e., a value for the type attribute) is always included, even if it

```
make-referring-expression($r$, $C$,$P$)
$L \leftarrow \{\}$
$D \leftarrow C$
for each member $A_i$ of list $P$ do
  $V$ = find-best-value($A_i$, basic-level-value($r$, $A_i$))
  if $V \neq$ nil $\wedge$ rules-out($\langle A_i, V\rangle$) $\neq$ nil
  then $L \leftarrow L \cup \{\langle A_i, V\rangle\}$
       $D \leftarrow D$ - rules-out($\langle A_i, V\rangle$)
  endif
  if $D = \{\}$ then
    if $\langle$type, $X\rangle \in L$ for some $X$
      then return $L$
      else return $L \cup \{\langle$type, basic-level-value($r$, type)$\rangle\}$
      endif
  endif
next
return failure

find-best-value($A$, initial-value)
if user-knows($r$, $\langle A$, initial-value$\rangle$) = true
then value $\leftarrow$ initial-value
else value $\leftarrow$ nil
endif
for $v_i \in$ taxonomy-children(initial-value)
  if $v_i$ subsumes value($r$, $A$) $\wedge$
    (new-value $\leftarrow$ find-best-value($A$, $v_i$)) $\neq$ nil $\wedge$
    (value = nil $\vee$
     |rules-out($\langle A$, new-value$\rangle$)| > |rules-out($\langle A$, value$\rangle$)|)
  then value $\leftarrow$ new-value
  endif
next
return value

rules-out($\langle A, V\rangle$)
return $\{x : x \in D \wedge$ user-knows($x$, $\langle A, V\rangle$) = false$\}$
```

Figure 1: The Algorithm

has no discriminatory power (in which case the basic level value is used); other attribute values are only included if, at the time they were under consideration, they had some discriminatory power.

More precisely, the algorithm is as shown in Figure 1. Here, $r$ is the intended referent, $C$ is the contrast set, $P$ is the list of preferred attributes, $D$ is the set of distractors (contrast set members) that have not yet been ruled out, and $L$ is the list of attribute–value pairs returned.[3]

make-referring-expression is the top level function. This returns a list of attribute–value pairs that specify a referring expression for the intended referent. Note that the attributes are tried in the order specified in the *preferred-attributes* list, and that a value for type is always included, even if type has no discriminatory power.

find-best-value takes an attribute and an initial value; it returns a value for that attribute that is subsumed by the initial value, accurately describes the intended referent (i.e., subsumes the value the intended referent possesses for the attribute), rules out as many distractors as possible, and, subject to these constraints, is as close as possible in the taxonomy to the initial value.

rules-out takes an attribute–value pair and returns the elements of the set of remaining distractors that are ruled out by this attribute–value pair.

## An Example

Assume the task is to create a referring expression for Object1 in a context that also includes Object2 and Object3:

- Object1: ⟨type, chihuahua⟩, ⟨size, small⟩, ⟨colour, black⟩
- Object2: ⟨type, chihuahua⟩, ⟨size, large⟩, ⟨colour, white⟩
- Object3: ⟨type, siamese-cat⟩, ⟨size, small⟩, ⟨colour, black⟩

In other words, $r$ = Object1 and $C$ = {Object2, Object3}. Assume that $P$ = {type, colour, size, ...}.

When make-referring-expression is called in this context, it initializes $L$ to the empty set and $D$ to $C$, i.e., to {Object2, Object3}. Find-best-value is then called with $A$ = type, and *initial-value* set to the basic-level type of Object1, which, let us assume, is dog.

Assume user-knows(Object1, ⟨type, dog⟩) is true, i.e., the user knows or can easily perceive that Object1 is a dog. Find-best-value then sets *value* to dog, and examines the taxonomic descendants of dog to see if any of them are accurate descriptions of Object1 (this is the subsumption test) and rule out more distractors than dog does. In this case, the only accurate child of dog is chihuahua, but ⟨type, chihuahua⟩ does not have more discriminatory power than ⟨type, dog⟩ (both rule out {Object3}), so find-best-value returns dog as the best value for the type attribute. Make-referring-expression then verifies that ⟨type, dog⟩ rules out at least one distractor, and therefore adds this attribute-value pair to $L$, while removing rules-out(⟨type, dog⟩) = {Object3} from $D$.

This means that the only remaining distractor in $D$ is Object2. Make-referring-expression (after checking that $D$ is not empty) calls find-best-value again with $A$ = colour (the second member of $P$). Find-best-value returns Object1's basic-level colour value, which is black, since no more specific colour term has more discriminatory power. Make-referring-expression then adds ⟨colour, black⟩ to $L$ and removes rules-out(⟨colour, black⟩) = {Object2} from $D$. $D$ is then empty, so the generation task is completed,

[3] For simplicity of exposition, the algorithm as described here returns failure if it is not possible to rule out all the members of the contrast set. A more robust algorithm might attempt to pursue other strategies here, e.g., generating a referring expression of the form *one of the Xs*, or modifying the contrast set by adding navigation information (navigation is discussed in the section on Future Work).

and make-referring-expression returns {(type, dog), (colour, black)} , i.e., a specification for the referring expression *the black dog*. Note that if $P$ had been {type, size, colour, ...} instead of {type, colour, size, ...}, make-referring-expression would have returned {(type, dog), (size, small)} instead, i.e., *the small dog*.

### Implementation

The algorithm is currently being used within the IDAS system [RML92]. IDAS is a natural language generation system that generates on-line documentation and help texts from a domain and linguistic knowledge base, using user expertise models, user task models, and discourse models.

IDAS uses a KL-ONE type knowledge representation system, with roles corresponding to attributes and fillers to values. The type attribute is implicit in the position of an object in the taxonomy, and is not explicitly represented. The value and taxonomy-children functions are defined in terms of standard knowledge-base access functions.

A knowledge-base author can specify explicit basic-level attribute values in IDAS user models, but IDAS is also capable of using heuristics to guess which value is basic-level. The heuristics are fairly simple (e.g., "use the most general value that is not in the upper-model [BKMW90] and has a one-word realization"), but they seem (so far) to be at least somewhat effective. A *preferred-attributes* list has been created for IDAS's domain (complex electronic machinery) by visual inspection of the equipment being documented; its first members are type, colour, and label. The user-knows function simply returns true if the attribute–value pair is accurate and false otherwise; this essentially assumes that the user can visually perceive the value of any attribute in *preferred-attributes*, which may not be true in general.

The referring expression generation model seems reasonably successful in IDAS. In particular, the algorithm has proven to be useful because:

1. It is fast. The algorithm runs in linear time in the number of distractors, which is probably impossible for any algorithm that includes an explicit brevity requirement (e.g., the algorithms of [Dal89,Rei90a]). Of equal importance, its run-time is independent of the number of potential attributes that could be used in the referring expression. This is a consequence of the fact that the algorithm does *not* attempt to find the attribute with the highest discriminatory power, but rather simply takes attributes from the *preferred-attributes* list until it has built a successful referring expression.
2. It allows human preferences and capabilities to be taken into consideration. The *preferred-attributes* list, the preference for basic-level values, and the user-knows function are all ways of biasing the algorithm towards generating referring expressions that use attributes and values that human hearers, with all their perceptual limitations, find easy to process.

Almost all referring expressions generated by IDAS contain a head noun and zero, one, or perhaps at most two modifiers; longer referring expressions are rare. The most important task of the algorithm is therefore to quickly generate easy-to-understand referring expressions in such simple cases; optimal handling of more complex referring expressions is less important, although the algorithm should be robust enough to generate something plausible if a long referring expression is needed.

## Future Work

### Navigation

As mentioned in the introduction, the algorithm presented here assumes that the intended referent is in the context set. An important question we need to address is what action should be taken if this is not the case, i.e., if the intended referent is not in the current focus of attention.

Unfortunately, we have very little data available on which to base a model of the generation of such referring expressions. Psycholinguistic researchers seem to have paid relatively little attention to such cases, and the transcripts we have (to date) examined have contained relatively few instances where the intended referent was not already salient.

However, we take the view that, in the general case, a referring expression contains two kinds of information: **navigation** and **discrimination**. Each descriptor used in a referring expression plays one of these two roles.

- Navigational, or **attention-directing** information, is intended to bring the intended referent into the hearer's focus of attention.
- Discrimination information is intended to distinguish the intended referent from other objects in the hearer's focus of attention; such information has been the subject of this paper.

Navigational information is not needed if the intended referent is already in the focus of attention. If it is needed, it frequently (although not always) takes the form of locational information. The IDAS system, for example, can generate referring expressions such as *the black power supply in the equipment rack*. In this case, *in the equipment rack* is navigation information that is intended to bring the equipment rack and its components into the hearer's focus of attention, while *black power supply* is discrimination information that is intended to distinguish the intended referent from other members of the context set (e.g., the white power supply that is also present in the equipment rack).

The navigation model currently implemented in IDAS is simplistic and not theoretically well-justified. We hope to do further research on building a better-justified model of navigation.

### Relative Attribute Values

As mentioned previously, the transcript analysis shows that human speakers and hearers often prefer relative instead of absolute attribute values, e.g., *small* instead of *one inch*. Knowledge bases sometimes explicitly encode relative attribute values (e.g., ⟨size, small⟩), but this can cause difficulties when referring expressions need to be generated in different contexts; a one-inch screw, for example, might be considered to be small in a context where the other screws were all two-inch screws, but large in a context where the other screws were all half-inch screws.

A better solution is for the knowledge base to record absolute attribute values, and then for the generation algorithm to automatically convert absolute values to relative values, depending on the values that other members of the context set possess for this attribute. Thus, the knowledge base might record that a particular screw had ⟨size, one-inch⟩, and the generation system would choose to call this screw *small* or *large* depending on the size of the other screws in the context set. We hope to do further research on determining how exactly this process should work.

## Conclusions

We have presented an algorithm for the generation of referring expressions that is substantially simpler and faster than the algorithms we have proposed in previous work [Dal89,Rei90a], largely because it performs much less length-oriented optimization of its output. We have been guided in this simplification effort by psycholinguistic findings and transcript analyses, and believe that the resulting algorithm is a more practical one for natural language generation systems than the ones we proposed previously.

## References

[App85] Douglas E Appelt. *Planning English Sentences.* Cambridge University Press, New York, 1985.

[BKMW90] John Bateman, Robert T Kasper, Johanna D Moore and Richard A Whitney. A General Organization of Knowledge for Natural Language Processing: The Penman Upper Model. Unpublished technical report, Information Sciences Institute/University of Southern California, 1990.

[BS85] Ronald Brachman and James Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9:171–216, 1985.

[Cru77] D. Cruse. The pragmatics of lexical specificity. *Journal of Linguistics*, 13:153–164, 1977.

[Dal88] Robert Dale. *Generating Referring Expressions in a Domain of Objects and Processes.* PhD Thesis, Centre for Cognitive Science, University of Edinburgh, 1988.

[Dal89] Robert Dale. Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 68–75. 1989.

[DH91] Robert Dale and Nicholas Haddock. Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4), 1991.

[Don66] Keith Donnellan. Reference and definite description. *Philosophical Review*, 75:281–304, 1966.

[FO75] William Ford and David Olson The elaboration of the noun phrase in children's description of objects. *Journal of Experimental Child Psychology*, 19:371–382, 1975.

[GJW83] Barbara Grosz, Aravind Joshi, and Scott Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 44–50. 1983.

[Gri75] H. Paul Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics: Vol 3, Speech Acts*, pages 43–58. Academic Press, New York, 1975.

[GS86] Barbara Grosz and Candace Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12:175–206, 1986.

[Kas89] Robert Kasper. A flexible interface for linking applications to Penman's sentence generator. *Proceedings of the 1989 DARPA Speech and Natural Language Workshop*, pages 153–158.

[Kro86] Amichai Kronfeld. Donnellan's distinction and a computational model of reference. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 186–191. 1986.

[Lev89] Willem Levelt. *Speaking: From Intention to Articulation.* MIT Press, 1989.

[Pec89] Thomas Pechmann. Incremental speech production and referential overspecification. *Linguistics*, 27:89–110, 1989.

[Rei90a] Ehud Reiter. The computational complexity of avoiding conversational implicatures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 97–104. 1990.

[Rei90b] Ehud Reiter. *Generating Appropriate Natural Language Object Descriptions.* PhD thesis, Aiken Computation Lab, Harvard University, 1990. Also available as Aiken Computation Lab technical report TR-10-90.

[Rei91] Ehud Reiter. A new model of lexical choice for nouns. *Computational Intelligence*, 7(4), 1991.

[RML92] Ehud Reiter, Chris Mellish, and John Levine. Automatic generation of on-line documentation in the IDAS project. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 64–71. 1992.

[Ros78] Eleanor Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum, Hillsdale, NJ, 1978.

[Sid81] Candace Sidner. Focusing in the comprehension of definite anaphora. In M. Brady and R. Berwick, editors, *Computational Models of Discourse*, pages 267–330. MIT Press, Cambridge, Mass, 1981.

[Son85] Susan Sonnenschein. The development of referential communication skills: Some situations in which speakers give redundant messages. *Journal of Psycholinguistic Research*, 14:489–508, 1985.

[Whi76] Grover Whitehurst. The development of communication: Changes with age and modeling. *Child Development*, 47:473–482, 1876.