Generating referring expressions with a unification grammar

Richard Power
Information Technology Research Institute
University of Brighton
Lewes Road
Brighton BN2 4AT, UK
Richard.Power@itri.bton.ac.uk

Abstract

A simple formalism is proposed to represent the contexts in which pronouns, definite/indefinite descriptions, and ordinal descriptions (e.g. 'the second book') can be used, and the way in which these expressions change the context. It is shown that referring expressions can be generated by a unification grammar provided that some phrase-structure rules are specially tailored to express entities in the current knowledge base.

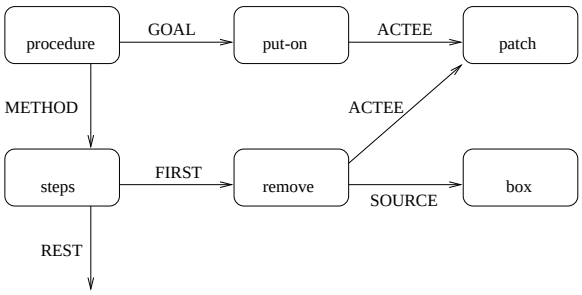

Figure 1: Network representation of an instruction

1 Introduction

Nominal referring expressions are exceptionally sensitive to linguistic context. If a discourse mentions a book, potential referring expressions include 'it', 'a book', 'the book', 'another book', 'the second book', along with an unlimited number of more complex descriptions (e.g. 'the red book') that mention the book's properties. The choice among these alternatives depends on features of the preceding text: whether the referent has been mentioned before; whether it is currently a focus of attention; whether different referents of the same type (e.g. other books) have been introduced as well. Taking account of such factors poses a tricky problem for Natural Language Generation (NLG), especially in applications in which efficiency (i.e. fast generation of texts) is a priority.

This paper proposes a method that allows efficient generation of referring expressions, through a unification grammar, at the cost of some initial effort in tailoring the phrase-structure rules to the current knowledge base. The method was invented to meet the needs of applications using 'WYSIWYM editing' (Power and Scott, 1998), which allow an author to control the content of an automatically generated text without prior training in knowledge engineering. WYSIWYM is based on the idea of a 'feedback text', i.e. a text, generated by the system, that presents the current content of the knowledge base (however incomplete) along with the set of permitted operations for extending or otherwise editing the knowledge; these operations are provided through pop-up menus which open on spans of the feedback text. Two requirements of WYSIWYM editing are that feedback texts should be generated fast (even a delay of a few seconds is irritating), and that they should express coreference relations clearly through appropriate referring expressions; reconciling these two requirements has motivated the work described here.

The semantic network in figure 1 shows a knowledge base that might be produced using the ICONOCLAST[1] system, which generates patient information leaflets. At present this knowledge base defines only the goal and first step of a procedure; before generating a useful output text the author would have to add further steps. To facilitate the author's task, the program generates the following feedback text, including the 'anchor' *Further steps* which provides options for extending the procedure.

[1]ICONOCLAST is supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant L77102.

To put on a patch:
1. Remove the patch from the box.
Further steps.

The program can also produce an 'output text' in which optional unspecified material is omitted. Whereas the feedback text is viewed only by the author during editing, the output text constitutes the final product which will be incorporated into the patient information leaflet. At the stage depicted by figure 1, with only one step specified, an output text could be generated if desired, but owing to the incomplete content it would read strangely:

Put on a patch by removing it from the box.

These simple texts already illustrate several ways in which the choice of referring expression depends upon context.

- To introduce a referent into the discourse, an indefinite description (e.g. 'a patch') is usually used, although a definite description may be preferred if the referent will already be familiar to the reader ('the box').

- Subsequent mentions of the referent are made through a pronoun or a definite description. In this way, the text distinguishes references to the same token from references to two tokens of the same type. If the patch removed from the box were different from the patch to be put on, the second line of the feedback text should contain another indefinite nominal (e.g. 'Remove a second patch from the box').

- Roughly, a pronoun can be used instead of a definite description if there is no danger of ambiguity, and if no major structural boundary has been passed since the referent was last mentioned. We are not concerned here with the details of this issue (Hofmann, 1989; Walker et al., 1998); in the examples, we have treated the colon in the feedback text as a major structural boundary, so preferring a definite description in the feedback text and a pronoun in the output text.

We concentrate here on two contextual features, **focus** and **prior mentions**. The problem of finding suitable identifying properties (Dale and Reiter, 1995; Horacek, 1997) will not be addressed here, although as will be shown our approach could incorporate this work.

2 Representing linguistic context

For any referring expression (e.g. 'a patch') one can define two relevant contextual states: first, the context in which the expression may be used; secondly, the context that results from its use. These will be called the 'initial' and 'final' contexts. In the case of 'a patch', they can be informally defined as follows.

Initial context: The patch is not in focus, it has not been mentioned before, and no other patch has been mentioned.

Final context: The patch is in focus, it has been mentioned, and no other patch has been mentioned.

The aim of this section is to model the initial and final contexts formally, considering not just indefinite descriptions but the full range of nominals mentioned earlier (including pronouns, definite descriptions and ordinal descriptions). For this purpose we will discuss an example that includes at least one nominal of each kind.

To put on a patch:
1. Take a sachet.
2. Remove the patch from a second sachet.
3. Position the patch and press it firmly.

The strange second step suggests that the author has made a mistake during knowledge editing, introducing a second sachet instead of re-using the sachet entity introduced in step 1. An important objective of the WYSIWYM feedback text is to expose such errors clearly. Because of this editing mistake, the passage mentions three objects: one patch, and two sachets. The patch is unique, the only object in the discourse satisfying the description 'patch'. The sachets, instead, are *distractors* — i.e., distinct objects answering to the same description.

As a first approximation, the contextual state can be formalized by two *vectors* which will be called the 'focus vector' and the 'mention vector'. Each vector should contain one element for each discourse referent that might be expressed by a nominal referring expression, so that in the example the vectors will be three elements long. The order of elements in the vector is irrelevant provided that it is observed consistently: it will be assumed arbitrarily that it is s_A, s_B, p, where s_A and s_B denote the two sachets and p denotes the patch. Note in particular that the order of s_A and s_B in the vector is independent from their order of introduction in the text.

The values in the focus vector are boolean: 1 if the referent is in focus, 0 if it is not. We simplify

by assuming (a) that focus is all-or-none rather than a matter of degree, and (b) that at most one referent can be in focus at any time. Actually the ICONOCLAST system refines the second limitation by grouping the referents according to whether they are competitors for the same pronoun: people compete for 'he/she' (or 'him/her' etc.), and physical objects for 'it'. With this refinement, the relevant constraint is that at most one referent in each group can be in focus at any time. However, in the example, the three referents are all physical objects — competitors for 'it' — so this complication can be ignored.

The behaviour of the focus vector is straightforward. At the beginning of the text no referent has been mentioned, so all focus values are zero:

	s_A	s_B	p
FOCUS	0	0	0

Whenever an object is mentioned, it comes into focus and its rivals go out of focus. As a result, the phrase 'the patch' in the final step switches the focus vector to the following:

	s_A	s_B	p
FOCUS	0	0	1

With p now in focus, the pronoun 'it' can be employed to refer to p in the final clause.

The mention vector is more complex. Each value is a ratio N/D, where N is the order of introduction of the referent relative to its distractors, and D is the number of members of the distractor group introduced so far. If the referent has not yet been mentioned, $N = 0$; if no members of the distractor group have yet been mentioned, $D = 0$. Initially all mention ratios are set to 0/0; at the end of step 1 in the example the state of the mention vector will be as follows (assuming that the first-mentioned sachet is s_A):

	s_A	s_B	p
MENTION	1/1	0/1	1/1

Consequently, when s_B is introduced during the second step, its initial mention ratio is 0/1, meaning that while s_B has not yet been mentioned, one of its distractors has got in first. On the basis of this information the generator should produce an indefinite description including the ordinal 'second' (or perhaps the determiner 'another'). By the end of step 2 all three objects have been introduced, so the mention vector reaches its final state:

	s_A	s_B	p
MENTION	1/2	2/2	1/1

Note that the two mentions of the patch in step 3 have no effect on the mention vector: its purpose is to record the order of introduction of a referent in relation to its distractors, not the number of times that a referent has been mentioned. When choosing a referring expression it is relevant *whether* a referent has been mentioned (as signalled by its N value in the mention ratio), but the precise number of mentions is of no significance.

It has been shown that the focus and mention vectors allow us to represent the initial and final contexts of the referring expressions in the example. (Of course we have oversimplified, especially in our treatment of focus.) We now show that by abstracting from the particular contexts in the example, it is possible to describe the initial and final contexts of these referring expressions in *all* texts expressing the same content. This is done by using variables to represent the values of any contextual features that do not interact with the referring expression under consideration. For instance, the generalized initial and final contexts of 'a patch' are

		Initial context			Final context		
p	'a patch'	s_A	s_B	p	s_A	s_B	p
	FOCUS	F_A	F_B	0	0	0	1
	MENTION	M_A	M_B	0/0	M_A	M_B	1/1

where F_A, M_A, etc. are variables. Among other things this rule implies that 'a patch' may be used whatever the current focus values for s_A and s_B, but that after 'a patch' these objects must be out of focus. Here are the corresponding rules for the other referring expressions in the example.

		Initial context			Final context		
p	'the patch'	s_A	s_B	p	s_A	s_B	p
	FOCUS	F_A	F_B	0	0	0	1
	MENTION	M_A	M_B	1/1	M_A	M_B	1/1

		Initial context			Final context		
p	'it'	s_A	s_B	p	s_A	s_B	p
	FOCUS	0	0	1	0	0	1
	MENTION	M_A	M_B	M	M_A	M_B	M

		Initial context			Final context		
s_A	'a sachet'	s_A	s_B	p	s_A	s_B	p
	FOCUS	0	0	F	1	0	0
	MENTION	0/0	0/0	M	1/1	0/1	M

		Initial context			Final context		
s_B	'a second sachet'	s_A	s_B	p	s_A	s_B	p
	FOCUS	F_A	0	F	0	1	0
	MENTION	1/1	0/1	M	1/2	2/2	M

Note that each rule is specific to a referent. For instance, the rule given for 'a sachet' is specific to s_A; a slightly different rule would be needed to describe the contexts in which 'a sachet' can be employed to refer to s_B.

3 Incorporating context into the grammar

A requirement on all WYSIWYM systems has been fast response. Every time that the author selects an editing operation on the feedback text, the knowledge base is updated and a new feedback text is generated. Any tangible delay in presenting the updated feedback text is irritating.

In pursuit of efficiency, ICONOCLAST employs a top-down generator coupled with a unification grammar. The grammar adheres strictly to Occam's razor: features or rules are admitted only if they contribute to generating the desired texts. ICONOCLAST is implemented in ProFIT (Erbach, 1995), so that feature structures are represented by Prolog terms and can be unified efficiently through Prolog term unification.

How can linguistic context be fitted into such a scheme? Ideally we would like to incorporate context into the phrase-structure rules, so that for example a rule introducing a pronoun would be applied only if the referent to be expressed had a value of 1 in the focus vector. Unfortunately such a rule could not be formulated in general terms: both its semantic features and its focus and mention vectors would depend on particular properties of the current knowledge base. However, nothing prevents us from constructing 'bespoke' rules, tailored to the current state of the knowledge base, every time that it is updated. At first sight this might seem a ridiculous waste of time — one would have to envisage beforehand all the ways in which every referent might be expressed — but in compensation the search phase of generation can proceed much faster, since all calculations relating to linguistic context have already been performed, and there is no danger that they might be duplicated.

Returning to the example in the previous section, let us work out the bespoke phrase-structure rules that should be added to the grammar so that it can refer to s_A, s_B and p. At this stage we do not know the exact contexts in which these referents will be introduced; these will depend on text-planning decisions during generation. Nevertheless, some valid generalizations can be made in advance by examining the content to be expressed:

- p will be mentioned several times, so we might need pronouns, definite descriptions, and indefinite descriptions. However, since p has no distractors, no rule introducing ordinals will be necessary.
- s_A and s_B are mentioned only once each, so definite descriptions and pronouns are unnecessary. However, since they are distractors, indefinite descriptions with ordinals should be provided.

Here is a phrase-structure rule generating indefinite descriptions for s_A (either 'a sachet' or 'a second sachet'). The rule is presented in simplified ProFIT notation, where `F!V` means that `V` is the value of feature `F`; as usual in Prolog, symbols starting with a lower-case letter are constants, while symbols starting with an upper-case letter are variables. Focus and mention vectors are represented by lists, while the phrase-structure constituents are listed under the `cset` feature. It will be seen that the rule does not rely entirely on unification, because it includes a statement expressing `Df` as a function of `Di`, but it will shown later how this blemish can be removed.

```
rule(referent!sA &
     properties![type:patch] &
     syntax!np &
     initial!(focus![0, _, _] &
              mention![0/Di, N/Di, M]) &
     final!(focus![1, 0, 0] &
            mention![Df/Df, N/Df, M])
     cset![properties![type:indef] &
           syntax!det,

           properties![order:(Df/Df),
                       type:patch] &
           syntax!nbar]) :-
  Df is Di + 1.
```

The syntactic form of this rule is $NP \rightarrow DET + NBAR$, where the $NBAR$ can be expanded by $NBAR \rightarrow NOUN$ to yield 'a sachet', and by $NBAR \rightarrow ORDINAL + NBAR$ to yield 'a second sachet'. Which of these rules is applied will depend on the `order` property, which reproduces the final mention ratio — a ratio of 1/1 activates the former rule, while any other ratio activates the latter.

The above statement of the rule simplifies by specifying contextual features only on the parent. In this particular case the omission is harmless: since the sachets have no properties (apart from `type`), the $NBAR$ of the indefinite description cannot include any expression referring to other objects (e.g. 'a sachet containing a patch'). In general, however, subordinated nominals might modify the context, so the final context of the parent should depend partly on the final context of its last constituent. This requires two things: first, the context must be 'threaded' through the constituents; secondly, the relationship between the final contexts of the parent and the last constituent must be defined.

The procedure for threading contextual features is straightforward. Suppose the rule has the form $u_0 \rightarrow u_1 + u_2 ... + u_N$, and that the initial and final contexts of any unit u are $I(u)$ and $F(u)$. In all cases, the initial context of the parent should be unified with the initial context of the first daughter, so that $I(u_0) = I(u_1)$. The relationship between $I(u_1)$ and $F(u_1)$ will depend upon the rule that expands the first daughter, but the final context of any daughter should always be unified with the initial context of the next daughter, so that for example $F(u_1) = I(u_2)$. Moreover, for any rule that does not generate a referring expression, the final context of the last daughter can be unified with that of the parent, so that $F(u_N) = F(u_0)$. For referring expressions, instead, $F(u_0)$ usually differs from $F(u_N)$, because the end of a referring expression is the point where the linguistic context may be changed.

Thus to take account of subordinated referring expressions, a rule must specify the relationship between three contexts: $I(u_0)$, $F(u_N)$, and $F(u_0)$. A rule capable of expressing s_A by 'a sachet containing a patch' should represent these contexts as follows:

$I(u_0)$	s_A	s_B	p
FOCUS	0	...	...
MENTION	$0/D_i$	N/D_i	...
$F(u_N)$	s_A	s_B	p
FOCUS	...	...	F
MENTION	$0/D_i$	N/D_i	M
$F(u_0)$	s_A	s_B	p
FOCUS	1	0	0
MENTION	D_f/D_f	N/D_f	M

where $D_f = D_i + 1$.

Finally we return, as promised, to the problem of updating mention ratios by unification, without resorting to statements like `Df is Di + 1`. This can be done by replacing numbers with lists of the appropriate length, so that for example the ratio 0/2 is represented by the term

```
[]/[_, _]
```

With this convention, the relationship between the mention ratios of $F(u_N)$ and $F(u_0)$ can be stated without an accompanying numerical constraint:

	s_A	s_B	p
$F(u_N)$	[ ]/D	N/D	M
$F(u_0)$	[_ | D]/[_ | D]	N/[_ | D]	M

4 Conclusion

Two ideas have been suggested:

- The linguistic context relevant to choosing nominal referring expressions can be formalized, in part, by vectors giving focus values and mention ratios for all potential referents. These features can be threaded through the text structure during generation by assigning initial and final contexts to each textual unit.

- Since generation requires search through a space of possible structures, there is a danger that expensive computations of linguistic context will be repeated many times. This can be avoided by composing 'bespoke' phrase-structure rules, tailored to the entities currently in the knowledge base, before embarking on the search phase.

Note that the first proposal can be employed independently from the second, which is more speculative. However, we think that the idea of using specially tailored phrase-structure rules deserves consideration. Its applications are not limited to the generation of referring expressions. One aim of the ICONOCLAST project is to generate texts in a variety of house styles, where a 'style' embraces preferences regarding textual organization, wording, punctuation and layout. To cover a large range of styles, many patterns must be made available to the generator, even though only a fraction are relevant for a particular company and a particular knowledge base. Before commencing a search through this space of patterns, it is worth devoting some effort to refining the search space by filtering out irrelevant rules and perhaps merging rules that separately constrain linguistic and presentational features.

The efficiency of the approach suggested here is difficult to evaluate in general terms: it will depend on the nature of the alternative methods, and also on the size of the generated text. For larger texts, in which entities may be mentioned many times, the initial investment of effort in creating bespoke phrase-structure rules will obviously pay more dividends. However, before trying to evaluate this difficult trade-off, we feel the next step should be to ensure that the approach can be applied to a wider range of referring expressions (e.g. demonstratives, plurals), and that it can be extended to cover a more complex treatment of focus such as centering theory (Walker et al., 1998).

Although we have not addressed here the problem of selecting appropriate properties for use in

referential descriptions (Dale and Reiter, 1995), it is worth noting that since this selection depends on the current state of the knowledge base, it can also be performed before the search phase of generation, the results of the selection algorithm being saved in the form of additional bespoke rules.

References

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

G. Erbach. 1995. Profit: Prolog with features, inheritance and templates. In *Seventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 180–187, Dublin.

T. Hofmann. 1989. Paragraphs and anaphora. *Journal of Pragmatics*, 13:239–250.

H. Horacek. 1997. An algorithm for generating referential descriptions with flexible interfaces. In *35th Annual Meeting of the Association for Computational Linguistics*, pages 206–213, Madrid.

R. Power and D. Scott. 1998. Multilingual authoring using feedback texts. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, pages 1053–1059, Montreal, Canada.

M. Walker, A. Joshi, and E. Prince. 1998. *Centering theory in discourse*. Clarendon Press, Oxford.