Referring Via Document Parts

Ivandré Paraboni[1] and Kees van Deemter[2]

[1] Instituto de Ciências Matemáticas e de Computação – ICMC,
Universidade de São Paulo - USP
Av Trabalhador São-Carlense, 400, 13560-970 - São Carlos SP, Brazil
ivandre@icmc.usp.br
[2] Department of Computing Science, King's College, University of Aberdeen,
Aberdeen AB24 3UE, Scotland, UK
kvdeemte@csd.abdn.ac.uk

Abstract. Documents in a wide range of genres often contain references to their own sections, pictures etc. We call such referring expressions instances of *Document Deixis*. The present work focuses on the generation of Document Deixis in the context of a particular kind of natural language generation system in which these descriptions are not specified as part of the input, i.e., when it is up *to the system* to decide whether a reference is called for and, if so, which document entity it should refer to. We ask under what circumstances it is advantageous to describe domain objects in terms of the document parts where they are mentioned (as in "the insulin described in section 2"). We report on an experiment suggesting that such indirect descriptions are preferred by human readers whenever they cause the generated descriptions to be shorter than they would otherwise be.

1 Introduction

Document parts such as sections, subsections, pictures, paragraphs etc may be referred to for various purposes, for example to point to additional information on the current topic of the text, e.g., "see also section 7". References to document parts will often be *deictic*, in the sense that the realisation of the expression depends on the place in the document where the referring expression is uttered (e.g., "this section" versus "section 1.1."). Accordingly, we will call the references to parts of the *same* document instances of Document Deixis (DDX).

We are interested in a particular kind of DDX, which we have previously called *object-level* instances of Document Deixis [9]. These are usually part of a larger expression which refers to a domain entity. The entity in question may be concrete (e.g., the medicines in Example 1) or abstract (e.g., the advice in Example 2). In the corpora that we investigated – patient information leaflets [1] - references to abstract entities or sets of them are far more common.

Example 1 (…) if you are taking any of *the medicines e.g. antibiotics listed under the section "Are you taking other medicines?"* (Cilest).

Example 2 You will notice that *the advice contained in this leaflet* may vary depending on whether you are (…) (Neoral Oral Solution)

A. Gelbukh (Ed.): CICLing 2006, LNCS 3878, pp. 299 – 310, 2006.

Unlike most concrete domain entities (e.g., objects in the physical world), such abstract referents often lack a concise textual description, and perhaps for that reason are referred to via document parts. Interestingly, many of these document-deictic referents can be also discourse deictic referents [4,12].

Document-deictic descriptions are ubiquitously found in many document genres, making their generation a practically relevant research topic on its own right, in connection with a certain class of natural language generation (NLG) systems for aiding in the authoring of complex documents [2]. In a system of this kind, the user, or author, creates a document by specifying its content (and sometimes providing also a high-level specification of its form) leaving to the system the responsibility over linguistic form and layout of the document.

Different systems may require different degrees of participation from the author. In most cases, however, low-level decisions such as lexical choice and referring expression generation are not under the author's control. For example, the author may specify that a certain domain entity has a certain property `P`, but it is up to the system to determine whether the entity will be referred to via a proper name, a pronoun, or a definite description, perhaps quoting a document part as in the previous example. The reminder of this paper investigates under what circumstances object-level Document Deixis is useful.

2 Document Deixis as 'Ordinary' Definite Descriptions

In this section we discuss how a 'traditional' algorithm for generating descriptions of domain entities in a given NLG application can be extended to generate certain instances of DDX as well. More specifically, we discuss how to adapt the well-known Dale & Reiter Incremental algorithm [3] to the task of generating ordinary and DDX descriptions alike. Briefly, the Incremental algorithm takes as its input a set of domain entities (the intended referent `r` and the context from which `r` has to be told apart) and their referable properties (pairs attribute-value such as 'colour-black' or 'type-dog') and a (domain-dependent) list of preferred attributes `P` specifying the order in which the algorithm will attempt to add properties to the description under generation. Properties that are *restrictive* (i.e., those that help ruling out distractors) are selected one by one, until a unique description is obtained (e.g., "the black dog").

We will limit ourselves to restrictive uses of document-deictic information. In this case, DDX can be viewed as an abbreviation device not unlike discourse anaphora, employed to avoid the repetition of a long description which has already been introduced in the discourse. We will assume that the choice between an anaphoric reference and a full definite description has already been made [6] and that an opportunity for a description (potentially making use of DDX) has been identified.

The semantic function of a restrictive DDX does not differ from that of 'ordinary' referring expressions, namely to single out a domain entity. One obvious approach is to assume that these expressions are constructed in the same way as ordinary referring expressions, e.g., by using the Incremental algorithm. Thus, document parts can be viewed as document-deictic properties of some of the domain entities they describe. For example, a domain entity realised as a picture may have, besides all its domain properties, the *document-related* property of 'being described by' the picture.

For concreteness, we will consider the following example of document structure, which we call our *target document*. The document consists of two sections (1 and 2) divided into three subsections (A, B and C) each. Each of the six subsections has a unique and well-defined topic (an insulin product), i.e., each topic is 'described by' a different document part. Although not truly representative of the documents found in the PILs corpus (whose leaflets are far less structured and do not usually describe different medicines in such ordered fashion), this organisation represents the minimal complexity required for discussing the most interesting instances of cross-section DDX such as "the insulin described in subsection C".

The following is a simplified representation of the target document, in which the domain entities representing the topic of each section are shown in brackets.

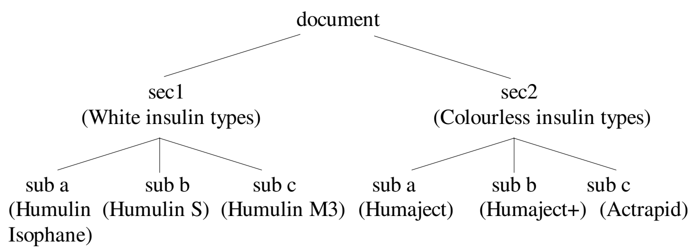


Fig. 1. Example of document structure

In this example, domain entities (i.e., different kinds of insulin products such as Humulin Isophane etc) have ordinary domain-related properties such as *type* (insulin), *colour* (white or colourless), *origin* (human or animal) etc. In addition to that, each domain entity has the document-related property of being *described_by* a particular document part. Thus, the description "the insulin mentioned in subsection C" may distinguish one product from all others, analogous to an 'ordinary' description like "the white animal insulin containing folic acid".

In order to avoid the problem of overlapping values (i.e., the problem of being described by section 1 and section 1a simultaneously, as reported in [10]) we assume that only the leaves of the document tree such as subsections (which correspond to the most specific values of the attribute) are modelled as the possible values of the *described_by* attribute. This will have also the effect of producing references whose referents are arguably easier to identify [8].

3 When to Generate Document-Deictic Descriptions

Object-level DDX represents a non-standard means of referring to domain entities, and perhaps for that reason these descriptions remain little investigated in the NLG field. In this section we discuss some circumstances under which it may be appropriate, or perhaps even necessary, to generate them.

Work on multimodal systems suggest a few possible triggers for reference to parts of a multimodal presentation. For example, in the COMET system [7] a reference such as “Remove the holding battery cover plate, highlighted in the right picture” can be generated when the system determines that the user does not know the term commonly used to describe an object depicted in the presentation. However, it is likely that a document-authoring system will have to take many other (linguistically-motivated) factors into account. For example, an entity may be referred to via a document part as in Example 3 simply because not referring to the document part would lead to a more complex description as in Example 4:

Example 3 Are you allergic to any of *the ingredients listed under section 5*?

Example 4 Are you allergic to diclofenac, mannitol, sodium metabisulphite (E223), benzyl alcohol, propylene glycol or sodium hydroxide?

We will focus on the cases in which document-related properties are viewed as a repair strategy: ordinary descriptions (i.e., those not using any document-related properties) are the method of choice, and document-related properties are used only to avoid overly long, or otherwise awkward descriptions, reflecting a Gricean-style brevity maxim [5]. Suppose, for example, the generator has to refer to a particular side effect, and that the description L produced by the Incremental algorithm makes use of the following domain-related properties:

‘being associated with medicine 1’
‘being associated with medicine 2’
‘being a long-term side effect’

Suppose that the side effect in question is the only one to be described in, say, section 1. In a case like this, the sheer length of L might be sufficient reason for the generation of DDX as in “The side effects described in section 1”, rather than “The long-term side effects that are associated with both medicine 1 and medicine 2”. In other words, the length of the description can be regarded as a factor for triggering instances of DDX.

Taking the Incremental algorithm as a basis, the idea of using description length as a factor to trigger the generation of DDX could be implemented in various ways. One such strategy would be varying the priority assigned to the *described_by* attribute, using it for example either before the inclusion of any domain property, or only after a number of domain properties have already been added. We call the former a *ddxFirst* strategy, and the latter *ddxLast*.

When *described_by* properties are highly discriminating (as in our target document), *ddxFirst* may produce fairly short descriptions such as “The insulin described in subsection A”. On the other hand, by regarding *described_by* as a last resort to disambiguate the reference, *ddxLast* may produce lengthy descriptions such as “The liquid insulin containing phenol described in subsection A of section 2”.

Using the length of the description as a trigger for DDX is comparable to the strategy adopted by van der Sluis [11] to generate *pointing gestures* for replacing overly long descriptions in the sense that both are strategies for facilitating reference. There is however one major difference between the two approaches: unlike pointing gestures in [11], document-related properties cannot be assumed to be uniquely distinguishing. After the inclusion of the *described_by* attribute it may be still necessary to include further (domain-related) properties to obtain a uniquely distinguishing description. By contrast, van der Sluis simply discards the generated description and generate a (uniquely distinguishing) pointing gesture instead.

The use of DDX as a means of rephrasing a 'bad' description can be implemented as follows. First, a description using only domain-related properties is attempted. If the length of the description rises above a certain level, then the description is discarded and a new reference is generated, this time using *described_by* as the most preferred attribute in the list P. We will call this the *regen(eration)* strategy.

The length of the description can be measured in different ways. We follow Dale & Reiter [3] in defining descriptions in terms of semantic rather than syntactic components, i.e., in terms of properties rather than words or other syntactic structures, even though the number of properties may not reflect the length of the surface string. This may be especially true for document-related properties. For example, a given set of symptoms referred to via a picture as in "the symptoms shown in pic.5" may convey only two attributes (*type* and *described_by*), depending on the surface realisation of the document-deictic description (which may require reference to more than one document entities this reference could be realised as "the symptoms shown in picture 3 in part B of section 2").

Both *ddxFirst* and *ddxLast* can be easily implemented as part of the Incremental algorithm by varying the position of the attribute *described_by* in the list of preferred attributes P. The *regen* strategy can be implemented as follows. In this representation, P is assumed to be the list of preferred attributes used by the *MakeReferringExpression* function, which is the core of the Dale & Reiter Incremental algorithm.

```
L ← MakeReferringExpression(r, C)
if (Length(L) > maxlength) or (L = ∅) then
      P ← <described_by> ∪ P
      L ← MakeReferringExpression(r, C)
return(L)
```

The following are examples of reference according to the above strategies (plus the *original* version, which does not make use of document-related properties). Note that the output of the *regen* strategy coincides with either *original* or (as in the example) *ddxFirst*. A list of descriptions generated by the four strategies in the context of our target document is presented in the appendix.

Table 1. Examples of descriptions produced by different generation strategies

Strategy	domain entity (insulin product)
original	"The animal liquid insulin containing phenol"
ddxFirst	"The insulin described in subsection A of section 2"
ddxLast	"The liquid insulin containing phenol described in subsection A of section 2"
regen	"The insulin described in subsection A of section 2"

To put into practice some of these ideas we implemented a simple DDX generator. The program displays four versions of our target document in which all the information except the referring expressions is canned text. In each version, the referring expressions are generated according to a different strategy that may include the use of document-related properties or not. In order to prevent the generation of an overly large number of these references, we assumed that an instance of DDX is only possible if (a) the reference is the first mention of the domain entity in the corresponding subsection and (b) in the case of reference to individual domain entities, the referent is described by a subsection other than the subsection containing the referring expression (i.e., a situation of non-local reference). We do not claim however any plausibility on these constraints. As we pointed out in the previous sections, adequacy constraints of this kind seem to be style- or genre-dependent, and our present choice is strongly based on the content and structure of the target document.

The generated documents are necessarily sketchy and repetitive, and their descriptions are sometimes clumsy owing to the large number of semantic properties conveyed (especially when not using document-related properties). Nevertheless, it is tempting to ask how the use of document-related properties affects their acceptability as referring expressions, as we will discuss in the next section.

4 An Experiment on the Use of Document Deixis

We carried out an experiment on the acceptability of selected instances of reference to individual domain entities generated according to the *original*, *ddxFirst*, *ddxLast* and *regen* strategies discussed above. Our goal was to find out (a) whether the predictions made by the *regen* strategy are accurate and (b) how the *ddxFirst* strategy compares to *ddxLast* and *original*. Our hypotheses were the following:

h.1a: when *regen* predicts *ddxFirst*, *ddxFirst* is preferred to *original*;

h.1b: when *regen* predicts *original*, *original* is preferred to *ddxFirst*;

h.2: *ddxFirst* is always preferred to *ddxLast*;

h.3: *ddxFirst* is always preferred to *original*.

Hypotheses h.1a and h.1b test the outcome of the *regen* strategy (which may produce descriptions making use of a document-related property or not). Hypothesis h.2 compares the strategy *ddxFirst* with *ddxLast*, and h.3 compares *ddxFirst* with *original*. Note that h.1b potentially contradicts h.3.

Method

Subjects were asked to rate the acceptability of instances of DDX generated by the implemented program described in the previous section.

Subjects: 20 graduate students.

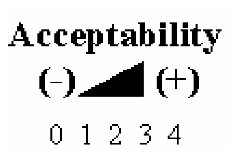
Materials: We chose to evaluate five contexts of reference (cf. appendix) found in the target document used in our implementation. Each context consisted of a different place of utterance and a different referent. In cases of DDX, the direction (backwards or forwards) and region (local or non-local) of reference were as evenly distributed as possible, although these factors were not expected to affect the results since the subjects did not have to identify the referents, but simply evaluate the wording of the referring expressions.

Because in some cases two different strategies produced the same output, it sufficed to test 14 descriptions covering the above cases, namely, five instances of *original*, five instances of *ddxFirst* and the instances of *ddxLast* in contexts 1-4. The evaluation of *ddxLast* in context 5 (which coincides with *original*) and the evaluation of *regen* (which coincides with *ddxFirst* in contexts 1-3, and with *original* in contexts 4-5) were simply derived from the evaluation of the main set of 14 descriptions.

The 14 references were presented in a printed format, keeping their original context (i.e., as in the target document used in the implementation). Minor changes were made in the format and content of the document (mainly involving text aggregation) to make it more appealing. In order to reduce the scope for comparison between different strategies that could lead to a bias against DDX (which seemed to occur in a preliminary pilot experiment), the three strategies of each situation of reference were split across three different versions of the same document (i.e., one document for each strategy, comprising five references in each of the first two documents and four references in the third one). The documents were identical except for the wording of the referring expressions.

Procedure: The subjects were presented a printed version (one page-long, with no page numbers) of the target document (called document 1) with no references to be rated, and they were asked questions about its content and form to guarantee that they were familiar with the setting of the experiment. Next, the subjects were given three versions of the same document (documents 2-4) containing the 14 references to be evaluated[1], and they were asked to judge each case on its own. The following is a fragment of one such question corresponding to *ddxFirst* in context 1 (underlined).

> This medicine is a liquid form of insulin. Because it contains phenol this medicine is not suitable for children under 12. The insulin described in subsection C is more appropriate for children under 12



The subjects rated the referring expressions from 0 (unacceptable) to 4 (highly acceptable). The greatest difference in results was obtained in the *ddxLast* strategy. Unlike *ddxFirst* and *original*, this strategy was never rated as highly acceptable, and it was the only one rated as unacceptable (11 cases, corresponding to 14% of the total). The *ddxLast* strategy fares better than the alternatives in only five cases (6%) and it fares worse in 40 (50%).

The results of the experiment are shown below. For each pair of strategies under consideration, we present the percentage of answers that favours each alternative (%) excluding the cases in which both alternatives were equally rated, the average score obtained by each strategy, the sum of ranks (w), degrees of freedom (N), and significance (p) obtained from the Wilcoxon's signed-rank test. All results are statistically significant as indicated. All hypotheses were confirmed except for h.3, for which there was an effect in the opposite direction.

[1] For the actual documents used in the experiment, see [9].

Table 2. Summary of the experiment results

	h.1a (p < .05)		h1.b (p < .005)	
	ddxFirst	*original*	*original*	*ddxFirst*
%	67.86%	32.14%	86.36%	13.64%
average	2.55	2.35	2.68	1.98
w / N	126.5 / 28		16.5 / 22	

	h.2 (p < .005)		h.3 (p < .05)	
	ddxFirst	*ddxLast*	*ddxFirst*	*original*
%	94.23%	5.77%	44.00%	56.00%
average	2.41	1.38	2.32	2.48
w / N	40.5 / 52		460 / 50	

Discussion

In contexts 1-3, in which the *regen* strategy predicts an instance of DDX (i.e., producing a description as in *ddxFirst*), *ddxFirst* was preferred to *original* in nearly 68% of the cases. This result confirms our hypothesis h.1a. Conversely, in contexts 4-5, in which the *regen* strategy does not predict *ddxFirst*, a description without document-related properties (as in the *original* strategy) is preferred to *ddxFirst* in 86% of the cases, hence confirming our hypothesis h.1b. Overall, the predictions of the *regen* strategy (triggering the generation of DDX when the description reaches a certain length) seem to be supported by these results.

The fact that the preference for *ddxFirst* in h.1a is smaller than the preference for *original* in h.1b may be related to the method used in the *regen* strategy to measure the length of the description. In the implemented algorithm, the length of the description is based on the number of existing (domain-related) properties. However, this method does not take into account the fact that the realisation of a document-related property may require a large linguistic expression. A single property of 'being described by' a certain document part may be realised for instance as "described in subsection B of section 2", which means that the *regen* strategy may in some cases favour a document-deictic expression whose surface realisation is actually longer than its non-deictic counterpart. While using the number of semantic properties to limit the length of the description may be convenient from the computational point of view, this method is unlikely to correspond to the criteria adopted by the subjects of the experiment to rate the descriptions, and perhaps for that reason some of them chose fewer *ddxFirst* references in the situations addressed by h.1a (or, conversely, more *original* references in h.1b). Although we do not attempt to prove this claim, we believe that the implementation of a more psychologically realistic method for measuring the length of the description (e.g., based on the number of words in the surface realisation) would result in a more accurate *regen* strategy.

The comparison between *ddxFirst* and *ddxLast* strategies in h.2 is straightforward. In 94% of the cases in which there was a difference between the two scores, *ddxFirst* fares better than *ddxLast*. This result is not surprising given the complexity of the descriptions produced by *ddxLast* and, accordingly, the poor rates obtained by this strategy in the experiment.

Finally, the overall comparison between *ddxFirst* and *original* strategies in h.3 shows that the *original* option was preferred in 56% of the cases. This result not only disconfirms hypothesis h.3, but shows an effect in the opposite direction. A possible explanation for this is discussed below.

Although the predictions of the *regen* strategy in h.1a and h.1b are supported by the experiment data, and although some conflict was to be expected (given that h.1b potentially contradicts h.3) the rejection of h.3 seems to suggest a simple link between the choice for DDX and the length of description. More specifically, there seems to be a preference for the *shortest* description in all the situations investigated regardless of the use of document-related properties. This appears to be the case in h.1a (which favours *ddxFirst* over *original*, being *ddxFirst* usually the shortest option), in h.1b (which favours *original* over *ddxFirst*, being *original* usually the shortest option) and in h.2 (which favours *ddxFirst* over *ddxLast*, being *ddxLast* the shortest option). Thus, we decided to check whether this observation could also explain the preference for *original* over *ddxFirst* in h.3. Our hypothesis was the following:

h.4: given the choice between *ddxFirst* with *original*, the shortest alternative is always preferred.

In order to test h.4, we first measured the length of all instances of *original* and *ddxFirst* descriptions in each of the five situations of reference, and determined the longest of the two. As an attempt to obtain a more accurate measurement, instead of taking the number of semantic properties into account (as in the *regen* strategy), we opted for measuring the length of the descriptions based on the number of words in their surface realisation. For this purpose, section labels and prepositions were also counted as words. For example, the length of the description "the insulin described in subsection A of section 2" according to this method is nine words. When two descriptions had the same length, we considered the *ddxFirst* alternative to be the shortest, the underlying assumption being that section labels (e.g., "A") and prepositions (e.g., "of") should count somehow less than e.g., nouns and adjectives.

Having determined the longest description of each situation of reference[2], we counted the number of cases in which these descriptions received the lowest score of the two options. Cases in which both options were rated as equal were eliminated from the analysis. Table 3 below summarises our findings obtained from the Wilcoxon's signed-rank test.

Table 3. Summary of the results for the additional hypothesis (h.4)

	h.4 ($p < .005$)	
	short	long
%	76.00%	24.00%
average	2.61	2.19
w / N	210 / 50	

[2] In contexts 1 and 3, the longest description was the *original* option, and in contexts 2, 4 and 5 it was the *ddxFirst* option.

In 76% of the cases in which there was a difference between the two scores, the shortest alternative (regardless of being produced by *original* or *ddxFirst*) fares significantly better as indicated. This observation is consistent with all our research hypotheses (including the low score obtained for *ddxLast*, since this strategy always produces the longest description of the three, and it fares worst in almost all cases). This also confirms the principles underlying the *regen* algorithm, suggesting that DDX can be used as a legitimate means of reducing the length of the description[3].

5 Final Remarks

We discussed the generation of document-deictic descriptions in systems in which these references are not originally planned as part of the input. In order to illustrate some of the issues under discussion, we implemented a simple Document Deixis generator, whose output was subsequently evaluated by subjects of an experiment.

The implementation work made use of an 'ideal' document structure in which the definition of 'describe' relations is straightforward. We did not address the question whether it is adequate to produce an instance of DDX in a particular place in the document. For example, a reference such as "See section B" is unlikely to be of much help if, say, it is overused throughout the text. What is considered overuse in one document genre may however be perfectly adequate in another. In our work these issues were large avoided by only allowing references to document parts under limited circumstances. The definition of well-justified constraints of this kind will however require further research.

We have discussed a number of triggers to DDX based on the length of the description under generation. We focused on the restrictive use of document-related properties and discussed how the Incremental algorithm could be adapted so as to use these properties alongside traditional, domain-related properties. Our experiment made use of a small number of referring expressions that were not based on naturally produced text. Despite these limitations, the results show a plausible link between description length and the use of document-related properties, suggesting that overly long description may in fact trigger the generation of Document Deixis in a way much similar to the generation of deictic pointing gestures.

Acknowledgements

This work was part of the PhD project of the first author and it has been supported by the CNPq – Brazilian research Council.

References

1. The Association of the British Pharmaceutical Industry: 1996-1997 ABPI Compendium of Patient Information Leaflets (1997)
2. Bouayad-Agha, N., et. al.: Integrating Content and Style in Documents: a Case Study of Patient Information Leaflets. Information Design Journal, 9 (2-3) (2000) 161-176

---

[3] This is not to say that shorter references are *always* better: for a discussion on when longer, logically-redundant descriptions may be preferred, see [8,9].

3. Dale, R. and E. Reiter: Computational interpretations of the Gricean maxims in the generation of referring expressions. Cognitive Science (19) (1995)
4. Eckert, M. and M. Strube: Resolving Discourse Deictic Anaphora in Dialogues. In Proceedings of EACL'99 Bergen (1999)
5. Grice, H. P.: Logic and Conversation. In P. Cole and J. L. Morgan (eds.) Syntax and Semantics, Vol. iii: Speech Acts. New York, Academic Press (1975) 41-58
6. McCoy, K. F. and M. Strube: Generating Anaphoric Expressions: Pronoun or Definite Description? In Proceedings of the ACL'99 Workshop "The Relation of Discourse/Dialogue Structure and Reference" University of Maryland (1999)
7. McKeown, K. et al.: Generating Cross-References for Multimedia Explanation. In Proceedings of AAAI-92 (1992) 9-16
8. Paraboni, I. and K. van Deemter: Generating Ease references: the case of document deixis. In Proceedings of the 2nd International Conference on Natural Language Generation INLG-2002, New York (2002) 113-119
9. Paraboni, I.: Generating References in Hierarchical Domains: the Case of Document Deixis. Information Technology Research Institute (ITRI), University of Brighton. Brighton, July (2003). A doctoral thesis. Also available as ITRI Technical Report ITRI-03-11 from http://www.itri.bton.ac.uk/.
10. van Deemter, K. : Generating Referring Expressions: Boolean Extensions of the Incremental Algorithm. In Computational Linguistics 28(1) (2002) 37-52
11. van der Sluis, I.: An Empirically Motivated Algorithm for Generation of Multimodal Referring Expressions. In Proceedings of the Student Research Workshop and Tutorial Abstracts of ACL 39th meeting (2001) Toulouse
12. Webber, B. L.: Structure and Ostension in the Interpretation of Discourse deixis. Language and Cognitive Processes 6(2) May (1991) 107-135

Appendix: Situations of Reference Considered in the Experiment

Context 1: forward local references in section 1 part A.

Original	*ddxFirst*	*ddxLast*	*Regen*
the white animal insulin containing folic acid	the insulin described in subsec C	the insulin containing folic acid described in subsec C	the insulin described in subsec C

Context 2: forward non-local references in section 1 part B.

original	*ddxFirst*	*ddxLast*	*regen*
the animal liquid insulin containing phenol	the insulin described in subsec A of section 2	the liquid insulin containing phenol described in subsec A of section 2	the insulin described in subsec A of section 2

Context 3: backward local references in section 1 part C.

original	*ddxFirst*	*ddxLast*	*regen*
the human liquid insulin containing phenol	the insulin described in subsec A	the liquid insulin containing phenol described in subsec A	the insulin described in subsec A

Context 4: backward non-local references in section 2 part A.			
original	*ddxFirst*	*ddxLast*	*regen*
the human insulin containing folic acid	the insulin described in subsec B of section 1	the insulin containing folic acid described in subsec B of section 1	the human insulin containing folic acid

Context 5: backward local references in section 2 part C.			
original	*ddxFirst*	*ddxLast*	*regen*
the soluble insulin containing phenol	the insulin described in subsec B	the soluble insulin containing phenol	the soluble insulin containing phenol