Context Modeling for Language and Speech Generation

Kees van Deemter
Philips Research Laboratories, WY 2.54
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
deemter@natlab.research.philips.com

1 Introduction

It is well known that some of the most important issues in the design of a dialogue system involve the modeling of linguistic context. The present paper highlights a number of these issues, focusing on the language and speech generation components of such systems, and discusses their implications for the way in which context has to be modeled in a spoken dialogue system. We will compare the 'dedicated' context models that have been proposed in theoretical and computational linguistics with the more general models proposed in artificial intelligence. Our main examples of a 'dedicated' context model will be the context model of the 'Dial Your Disc' (DYD) music information system (Collier and Landsbergen, 1995), (van Deemter and Odijk, 1997) and the better-known Discourse Representation Theory (e.g. (Kamp and Reyle, 1993)) of which this model is a variant. Our main example of a 'general' context model is provided by the so-called 'Ist' formalism (McCarthy, 1993).

2 A sketch of the dyd system

The DYD system produces spoken monologues derived from information stored in a general-purpose database about W.A.Mozart's instrumental compositions. The goal of the monologue generator is to generate from these data a large variety of spoken texts. A generator like this could be part of an electronic shopping system, where the system provides information and 'sales talk'. The way in which users can indicate their areas of interest will not be discussed in this paper, which focuses on language and speech generation. A (highly simplified) example of a database representation of a recording is:

```
[KV]        32
[DATE]      03/1766 - 04/1766
[SORT]      quodlibet
[TITLE]     Galimathias Musicum
```

A teleshopping system has to be entertaining. Therefore, an important system requirement is that a large variety of texts can be produced from the same database structures. Presentations are generated on the basis of database information by making use of syntactic sentence templates (Henceforth, S-template): structured sentences with variables, i.e., open slots for which expressions can be substituted. These syntactically structured sentence templates indicate how the information provided by a database object can be expressed in natural language. The required variety is achieved by having many different templates for the same information and by having a flexible mechanism for combining the generated sentences into texts. A template can be used, in principle, if there is enough information in the database to fill its slots. However, there are extra conditions to guard the well-formedness and effectiveness of presentations. For example, certain points in the discourse are more appropriate for the expression of a certain bit of information. Thus, it is important for the system to maintain a record showing which information has been expressed and when it has been expressed. This record, which is called the Knowledge State, will be part of DYD's Context Model.

Many variations of the above presentation are possible. The system can, for instance, start mentioning the date of composition, or information could be added that contrasts this composition with a previous one. Also, there are various ways of referring to the composition being discussed, for instance by name K. 309, with a definite noun phrase or with a pronoun. The appropriateness of a referring expression depends, among other things, on the existence and kinds of references to the referred object in previous sentences. Therefore, it is important to maintain a record of which objects have been introduced in the text, and how and when they have been referred to. This record will be called the Discourse Model, which is also a part of the Context Model.

As was mentioned above, templates in our system are structured sentences with slots for variable parts. For brevity, we will not represent syntactic structure but only the terminals of templates:

> (composition) was/were written by (composer) (date)

Slots are to be filled with structured expressions that contain database information. This is done with other, smaller, S-templates. The system has three modules: Generation, Prosody and Speech. The module Generation generates syntax trees on the basis of the Mozart database, a collection of S-templates, and the Context Model. Conversely, it updates the Context Model whenever a phrase has been generated. The module Prosody transforms a syntax tree into a sequence of annotated words, the annotations specifying accents and prosodic boundaries (e.g. pauses). The module Speech transforms a sequence of annotated words into a speech signal (Collier and Landsbergen, 1995).

3 Text Generation

As explained in the previous section, sentences are generated by means of S-templates. An S-template indicates how the meaning of a database record can be put into words. Given the information represented about the composition K.32 in the database, example sentences derived from the abovementiuoned S-templates include: *K.32 was written by W.A.Mozart in 1766* and *This quodlibet was written by the composer in March 1766.* The fact that S-templates are syntactically structured objects makes it possible to formulate various conditions on the form of variable parts. In this way, it is possible to avoid the generation of incorrect sentences such as *It were written by him when Mozart was only ten years old.* Since S-templates are structured objects, conditions guaranteeing the appropriate choice for the variable parts of the templates can refer to information contained in these structures. For instance, it can be read off the syntactic structure that the pronoun 'it' is the singular subject of the second sentence and that therefore the finite verb should be 'was'.

Which sentences should be used in a given situation? First, it has to be determined what is going to be said. This is determined during the dialogue, where the user can indicate a preference for less or more elaborate monologues. This preference is stored in the Dialogue State, a part of the Context Model in which all those properties of the dialogue history are recorded that are relevant for monologue generation.

Secondly, a selection has to be made from all S-templates in such a way that the text generated conveys all and only the required information. Only those S-templates are selected which are able to convey the relevant information; moreover, under normal circumstances, the same information is presented not more than once. These requirements have been incorporated in the text generator, which also presents the sentences in such a way that the text shows a certain coherence. Information should be grouped into convenient clusters and presented in a natural order. Clustering is achieved by means of the so-called Topic State. For each paragraph of the monologue, the Topic State, which is another part of the Context Model, keeps track of the topic of the paragraph, which is defined as a set of attributes from the (music) database. For example, a paragraph may have 'place and date of performance' as its topic and then only those S-templates can be used that are associated with the attributes 'date' and 'place'.

The text generator operates as follows: Each S-template 'attempts' to get a sentence generated from it into the text. Whether this succeeds depends on the information conveyed by the sentence, which information has been conveyed earlier, and whether the sentence can find a place in a natural grouping of sentences in paragraphs. Only local conditions on the Context Model and the properties of the current S-template determine whether a sentence is appropriate at a certain point in the text. No global properties of the text are considered and no explicit planning is involved.

As we have seen, an important part of the Context Model is a Discourse Model. Starting with an empty Discourse Model, each candidate sentence adds discourse referents and relevant associated information to this model. For example, the Discourse Model may record that a certain description (e.g., 'this composition') has occurred as the x-th and x+1-st word of the y-th sentence of paragraph number z of the u-th monologue that has occurred during a given user-system interaction. Rules for anaphora establish the antecedents for anaphora, and afterwards it is checked whether the resulting Discourse Model is well-formed (e.g., by checking whether each pronoun has an antecedent, whether definite descriptions have been used appropriately, etc.). If the Discourse Model is found to be well-formed, the candidate sentence can be used as an actual sentence. If not, a different candidate sentence is subjected to examination, etc. We will see that very similar rules, which are also based on the information in the Discourse Model, are used to determine which words in the sentence are to be accented.

4 Prosody and speech

Generating acceptable speech requires syntactic and semantic information that is hard to extract from unannotated text. In the present setting, however, speech generation is helped by the availability of syntactic and semantic information. When the generation module outputs a sentence, the generated structure contains all the syntactic information that was present in the S-template from which it results. Moreover, the Discourse Model, as we have seen, contains semantic information about the sentence. Both kinds of information are used to find the proper locations for pitch accents.

Existing speech synthesis systems (e.g., Bell Labs' Newspeak program) have typically de-stressed all content words that had occurred in the recent past. Yet, these systems still stress too many words (Hirschberg, 1990). To remedy this defect, we have redefined givenness and newness as properties not of individual words, but of entire phrases (van Deemter, 1994). These definitions are combined with a version of Focus-Accent theory to determine the exact word at which the accent must land.

Inspection of the relevant facts suggests strongly that words of very different forms may cause a word to have 'given' status. For example, an occurrence of 'K.32' or of 'this composition' may become 'given', and hence de-stressed (de-accented) due to an earlier reference to K.32:

> You have selected K.32.
> You will now hear K.32/this composition.

De-stressing and pronominalization occur in roughly the same environments, namely those in which an expression contains 'given' information. This suggests that both may be viewed as reduction phenomena that are caused by semantic redundancy (Van Deemter 1994). The Discourse Model presents itself as a natural candidate to implement this idea, since it contains all the relevant information. In particular, it says, for each referentially used Noun Phrase, whether and where in the discourse the object that it refers to was described earlier. If such an 'antecedent' for an expression is found earlier in the same paragraph, the expression is considered 'given' information (i.e., it is not 'in focus'). If not, it is considered 'new' (i.e., it is 'in focus').

The basic insight of Focus-Accent (e.g. (Ladd, 1980)) is the idea that the syntactic structure of a sentence determines its 'metrical' structure. Metrical structure is most conveniently represented by binary trees, in which one daughter of each node is marked as 'strong' and the other as 'weak'. Metrical structure determines which leaves of the tree are most suitable to carry an accent on syntactic grounds. Roughly, these are the leaves that can be reached through a path that starts from an expression that is 'in focus', and that does not contain weak nodes (Dirksen, 1992). More exactly, if a given major phrase is 'in focus', it is also marked as accented, and so is each strong node that is the daughter of a node that is marked as accented. Accent is realized on those leaves that are marked as accented. However, several obstacles may prevent this from happening. For example,

> (a) A major phrase is marked -A if it is not in focus.
> (b) A leaf x is marked -A if there is a recent occurrence of an expression y which is semantically subsumed by x.
> (c) A leaf is marked -A if it is lexically marked as unfit to carry an accent that is due to informational status. (Examples: 'the', 'a', some prepositions.)

The result of an -A marking is that the so-called Default Accent rule (cf. Ladd 1980) is triggered, which transforms one metrical tree into another:

> Default Accent rule: If a strong node n_1 is marked -A, while its weak sister n_2 is not, then the strong/weak labeling of the sisters is reversed: n_1 is now marked weak, and n_2 is marked strong.

In English, it is usually, but not always, the right daughter of a mother node that is strong. Thus, the metrical tree for our earlier example looks as follows:

```
   /    \
you,w  will now hear,s
           /   \
       will,w  now hear K.32,s
                   /   \
              now,w  hear K.32,s
                         /   \
                    hear,w  K.32,s
```

Assume that the Verb Phrase is 'in focus' and therefore labeled as accented. If semantic factors would not intervene, K.32 would carry an accent. But since K.32 is also referred to in the previous sentence of the discourse, K.32 represents 'given' information, and is marked -A. As a result, the Default Accent rule swaps the strong/weak (S/W) labeling between 'hear' and 'K.32' before the 'accented' labels are assigned. Consequently, the sentence accent trickles down along a path of strong nodes and ends up on 'hear'.

5 Context modeling

We have seen how the Knowledge State, the Topic State, the Context State, and the Dialogue State together form one large Context Model which is used (and maintained) by the DYD system to generate its spoken monologues. But context models have also come up in other settings. Wouldn't it have been possible to re-use these context models for our purposes?

5.1 Context Modeling in AI

One might try to use a general-purpose theory of context to formalize DYDs Context Model. The so-called 'Ist' theory (McCarthy, 1993), (Buvač, 1996) can be used for this purpose. Ist(c,p) can be read as saying that p is true with respect to c. Now let c be the context that obtains after the sentence 'Mozart composed K.280' has been generated. We can now say various things about c, and then use the Ist-formalism to say that a second sentence (for instance, 'It is a sonata') is expressed in c. The notation DE(c) stands for the set of 'discourse entities' (roughly: earlier-introduced individuals) associated with c:

Text(c) = Mozart composed K.280
Speaker(c) = dyd
Previous sentence(c) = ...
DE(c) = $\{x, y\}$
Conditions(c) =
{ x=W.A.Mozart,y=K.280,x composed y }
Ist(c,It is a sonata), *etc.*

The 'DE' predicate plays the role of DYD's so-called Discourse Model, noting which objects in the database have been referred to in the monologue. This information can be exploited when the second utterance, *It is a sonata*, is interpreted 'in the context of' *c*. This suggests that important parts of DYD's Context Model may be mirrored in the Ist-formalism. But *linguistic* contexts have a peculiarity: they change during processing: discourse entities are added, objects and expressions move into and out of focus as sentences are generated or interpreted. This requires extensions of the Ist formalism. For example, one need an 'update' operator '+' to say how a context *c* is changed when the sentence *S* has been processed in *c*:

$$c + S = c'$$

Also, one needs several operators to compare contexts. Thus, one might write

$$c[x, y]c'$$

to express that c and c' are alike, except for the discourse entities x and y. Using such extensions, Discourse Representation Theory can be mirrored in the Ist formalism. This is a useful exercise, which leads to a better understanding of the peculiarities of *linguistic* context. But it also raises the question of whether we might have used DRT as a backbone for DYD's Context Model.

5.2 Context Modeling in DRT

In the setting of DYD, DRT could take the form of a context model containing a series of sub-DRSs, the first of which contains information extracted from the dialogue that has led up to the selection of the first composition plus the monologue following it, and so on. However, setting up structures of this kind would have required a tremendous amount of work since generation requires many kinds of information that are neither routinely represented in existing versions of DRT, nor trivial to calculate on the basis of them. For example, DRSs do not normally contain a representation of their subject matter (their 'topic') and it would not be a trivial matter to deduce this information from the truth conditions of the DRS (Demolombe and Jones, 1995). Furthermore, standard versions of DRT do not contain information about the exact place of occurrence of expressions, nor do they contain information about paragraph structure. Of course, information of all these kinds might be added. The result would be a new, extended version of DRT, which would complicate drastically the formal basis of this theory (Muskens et al., 1996). Moreover, conventional DRSs contain plenty of semantic information that is not immediately relevant for current (i.e., generative) purposes. DRSs contain both less and more than what is needed for language generation.

The conclusion seems unavoidable: Language generation requires a specific kind of context models which are suitable to formalize the notion of a linguistic context. DYD's Context Model was designed to be such a context model. It might be viewed as a modest, computationally feasible version of DRT. This context model, with all its diverse components, may not be as *elegant* as some of the context models discussed in the present section. But it is difficult to see how the requirements of high-quality language and speech generation can be reconciled with formal elegance.

References

S. Buvač. 1996. Resolving lexical ambiguity using a formal theory of context. In K.van Deemter and S.Peters, editors, *Semantic Ambiguity and Underspecification.* CSLI Publications.

René Collier and Jan Landsbergen. 1995. Language and speech generation. *Philips Journal of Research*, 49(4):419–437.

Demolombe and Jones. 1995. Reasoning about topics: Towards a formal theory. In *Working Notes of Workshop on Formalizing Context.* AAAI.

A. Dirksen. 1992. Accenting and deaccenting: a declarative approach. In *Proc. of COLING*, Nantes, France.

J. Hirschberg. 1990. Accent and discourse context: assigning pitch accent in synthetic speech. In *Proc. of AAAI*, page 953.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic*, volume 42 of *Studies in Linguistics and Philosophy.* Kluwer Academic Publishers, Dordrecht.

D.R. Ladd. 1980. *The Structure of Intonational Meaning: Evidence from English.* Indiana University Press.

John McCarthy. 1993. Notes of formalizing context. In *Proceedings of IJCAI.*

Reinhard Muskens, Johan van Benthem, and Albert Visser. 1996. Dynamics. In J.van Benthem and A.ter Meulen, editors, *Handbook of Logic and Language.* Elsevier Science Publishers.

Kees van Deemter and Jan Odijk. 1997. Context modeling and the generation of spoken discourse. *Speech Communication*, 21(1/2):101–121.

K. van Deemter. 1994. What's new? A semantic perspective on sentence accent. *Journal of Semantics*, 11:1–31.