Dealing with distinguishing descriptions in a guided composition system

Pascal Mouret **Monique Rolbert**
Laboratoire d'Informatique de Marseille
Université de la Mediterranée
Faculte des Sciences de Luminy
Case 901, 163 Avenue de Luminy
13288 Marseille, France
{Pascal.Mouret,Monique.Rolbert}@lim.univ-mrs.fr}

Abstract: The goal of this paper is to provide computable account for some definite descriptions. To this end, we define in terms of inclusion the notion of distinguishing description and of distinguishable entities introduced by [Dale 89]. These definitions allow us to give conditions of wellformedness for *incomplete* distinguishing descriptions. We also extend the notion of distinguishing description to take into account cases of synonymy and hyponymy.

We describe a real application of a guided composition system where this sort of expressions arise.

1.Introduction

Many studies in natural language processing are concerned with how to generate or understand definite descriptions that evoke a discourse entity already introduced in the context. An interesting solution to this problem was proposed by [Dale 89] in terms of distinguishing descriptions and distinguishable entities.

Informally, a distinguishing description is a definite description which designates one and only one entity among others in a context set. Many natural language interfaces need to deal with this sort of NPs. We develop an application where these NPs occur in a particular way: the application works in a guided composition mode where correct sentences have to be produced word by word by the system. So, the problem of the (contextual) correctness of an *incomplete distinguishing description* arises. Moreover, the system has to understand (complete or incomplete) distinguishing descriptions in wider cases of references than those intended by [Dale 89], including cases of hyponymy and so on. We are going to show that the way we refine the notions introduced by Dale allows us to treat these two points in a rather simple and efficient manner.

In §2, we present our definitions of distinguishing descriptions and distinguishable entities and we use them to deal with incomplete distinguishing descriptions and wider cases of reference. An algorithm for incomplete distinguishing description production is given. §3 presents the application, §4 related works and §5 concludes.

2.Dealing with distinguishing descriptions

Following [Dale 96], let us consider that the context contains a set of entities $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ and that each entity e_i is described by the set of its properties $\mathcal{P}_{e_i}$.

[Dale 89] introduced the notion of *distinguishing description* (*dd*) which is the linguistic realisation of a set of properties which are together true of an entity e_i, but of no other entity in $\mathcal{E}$. [Dale 89] also mentions the notion of *distinguishable entity*, which is intuitively an entity that can be distinguished from the others by the use of a distinguishing description made from its set of properties.

Using inclusion between sets of properties, we can say that:

1. an entity e_i in $\mathcal{E}$ is distinguishable if there is no entity e_j in $\mathcal{E}$ such that $\mathcal{P}_{e_i} \subseteq \mathcal{P}_{e_j}$.

Similarly to the process of generating a *dd* from the set of properties of an entity to designate it, we will consider that to every definite description corresponds the set of properties (noted $\mathcal{P}_{dd}$) that it contains (informally the set of properties of which the definite description is the linguistic realisation). We will say that a definite description *designates* every entity e_i of $\mathcal{E}$ such that $\mathcal{P}_{dd} \subseteq \mathcal{P}_{e_i}$ (the entity *agrees* with the description).

Then, according to the definition given in [Dale 89] and the uniqueness requirement, we can say that:

2. a definite description is a distinguishing description if $\{e_i / \mathcal{P}_{dd} \subseteq \mathcal{P}_{e_i}\}$ is a singleton.

These two definitions will help us to deal with incomplete descriptions and to extend the notion of distinguishing description.

2.1Treating distinguishing descriptions in a guided composition system

Guided composition is a paradigm for NLP which is an answer to various limitations to NLP interfaces, especially limitations due to coverage of lexicons and grammars ([Rincel & al 89]). The basic idea is to inform the user, at every step, about the abilities of the system: for example, such a system can allow the user to ask what word(s) can appear at a time in a sentence. Then, the user chooses among the words proposed and so on. Therefore, the system must provide only expected words, i.e. words that can lead to a correct whole sentence.

In particular, the system must generate partially (word by word) definite descriptions which have to be correct from a contextual point of view when they form complete NPs. In a guided composition point of view, the problem is then to find how to know *as early as possible* if a string of words (an *incomplete distinguishing description* or *Idd*) may or may not lead to a correct distinguishing description.

In order to treat *Idd*, we have to define conditions so as to decide when it will lead to a correct definite description. For example, if there are two kings on a chessboard, one black and one white, the definite description 'le roi' ('the king') is a correct definite description from a syntactical point of view, but not from a contextual point of view because two entities of the context can be designated by it. However, it is a contextually correct *incomplete* definite description because it can lead to a correct distinguishing description if completed by 'noir' ('black') or 'blanc' ('white'). Conversely, if there is no bishop on the chessboard, the definite description 'le fou' ('the bishop') is neither a distinguishing description nor a (contextually) correct *incomplete dd* because it doesn't designate any entity of the context and, moreover, can't be completed to designate one.

So, an *Idd* is considered as correct if it can lead to a definite description which is a *dd*. As seen in this example, an *Idd* can designate more than one entity (as for the *dd*, we note $\mathcal{P}_{Idd}$ the set of properties from which an *Idd* is made and the *Idd* designates every entity e_i such that $\mathcal{P}_{Idd} \subseteq \mathcal{P}_{e_i}$); so it is clear that the uniqueness requirement is not adequate to caracterise correct *Idd*.

Let $\mathcal{D}_{\mathcal{E}}$ be the set of distinguishable entities of $\mathcal{E}$. In order to be sure that an *Idd* can always be completed to be a *dd*, a necessary and sufficient property is that some of the entities designated by the *Idd* are in $\mathcal{D}_{\mathcal{E}}$, i.e

3. an Idd is correct iff $\{e_i / \mathcal{P}_{Idd} \subseteq \mathcal{P}_{e_i}\} \cap \mathcal{D}_{\mathcal{E}} \neq \emptyset$.

Actually, an *Idd* which designates entities in $\mathcal{D}_{\mathcal{E}}$ can be continued into a *dd* by using properties which lead to designate one and only one of these entities.

Finally,

4. an *Idd* can be a *dd* if there exists one and only one entity e_i in $\mathcal{E}$ that agrees with it.

Notice that the uniqueness requirement must be met on $\mathcal{E}$ and not $\mathcal{D}_{\mathcal{E}}$. Actually, there may be only one entity in $\mathcal{D}_{\mathcal{E}}$ that agrees with an *Idd* and several others in $\mathcal{E}$ that agree with it (see example below). In this case, the *Idd* is correct but is not yet a *dd*. It is also interesting to note that the only entity that agrees with the *dd is* the referent of the definite description. That is to say that the process of verifying if an *Idd* is correct leads to solve the definite description in the end.

Let us have a look at an example:

Suppose we have a set of entities $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, the properties of which are :

e_1 is a dog,
e_2 is a dog that barks,
e_3 is a dog that barks,
e_4 is a red bird that sings,
e_5 is a red bird,
e_6 is a bird that flies.

In this example, e_2 and e_3 are not distinguishable, because their sets of properties are identical (and so $\mathcal{P}_{e_3} \subseteq \mathcal{P}_{e_2}$ and $\mathcal{P}_{e_2} \subseteq \mathcal{P}_{e_3}$). e_1 is not distinguishable because $\mathcal{P}_{e_1} \subseteq \mathcal{P}_{e_2}$. Finally, e_5 is not distinguishable because $\mathcal{P}_{e_5} \subseteq \mathcal{P}_{e_4}$. Therefore, only e_4 and e_6 are distinguishable entities. So, the set $\mathcal{D}_{\mathcal{E}}$ is not empty and, in the guided composition mode, the *Idd* 'le' ('the') can be proposed. What words can be proposed after? According to this context, 'le chien' ('the dog') is not a correct *Idd* because there is no distinguishable entity agreeing with it. On the contrary, 'l'oiseau' ('the bird') is correct and can lead to at least three different *dd*: 'l'oiseau qui chante' (the bird that sings') which designate e_4 and is minimal, 'l'oiseau rouge qui chante' ('the red bird that sings') which designates e_4 but is not minimal and 'l'oiseau qui vole' ('the bird that flies') which designates e_6.

Notice that, for instance, 'l'oiseau rouge' ('the red bird') is not a *dd*, although there is exactly one distinguishable entity (e_4) that agrees with it: this *dd* designates also e_5. The uniqueness requirement must be met on the whole set of entities from the context, and not only on the set of all distinguishable entities.

2.2 Refining the notion of distinguishing description

Using the definition of distinguishing description based on inclusion given above, we are going to show that this notion can be refined to take into account more subtle cases of reference. We define a more discerning inclusion between surface descriptions and sets of properties in order to treat cases of synonymy, nominalisation, hyponymy and so on: we want to generalize the notion of agreement to every case where a description is able to "evoke" an entity. Actually, it is well known (see for example [Corblin 95]) that an entity can be designated not only in terms strictly equal to those which served to introduce it. For example, a dog can be designated by 'the animal' or a child who robs something by 'the robber'.

To take these cases into account, we define the ~inclusion (noted $\tilde{\subseteq}$) between sets of properties as follow:

- some ~inclusion are given (these are the basic one) like for instance {animal} $\tilde{\subseteq}$ {dog} or {robber} $\tilde{\subseteq}$ {child, rob} i.e representations of relations of hyponymy, synonymy, and so on. These inclusions have to be known by the system;
- for all P, P', P'' sets of properties, if $P \tilde{\subseteq} P' \subseteq P''$ then $P \tilde{\subseteq} P''$;
- for all P, P' sets of properties, if P $\tilde{\subseteq}$ P', then P-P' can be partionned into subsets which are ~included in P'.

Basic ~inclusion representing relations of hyponymy is not too hard to define, and is useful for a lot of treatments in NLP (conceptual aspects of NLP, for example). Other basic ~inclusions need more knowledge (about the world and the language), not so easy to implement, but also necessary in other parts of the treatment of natural language.

The ~inclusion is transitive and works as expected on unions of sets. We use it (instead of simple inclusion) to compare a set of properties of an *Idd* (or a *dd*) and a set of properties of an entity and we say that the set of entities designated by a *dd* (or a *Idd*) is then $\{e_i / \mathcal{P}_{dd} \tilde{\subseteq} \mathcal{P}_{e_i}\}$.

So, this inclusion allows to use the distinguishing description 'the robber' to designate an entity that is a child who robs something. So it's a nice extension of the inclusion, when used to define the sets of entity an *Idd* (or a *dd*) designates.

It is also interesting to see if ~inclusion can be used between sets of entities' properties to define the notion of distinguishable entity. Suppose we have two entities in the context, one which is a robber and the other which is a child who has robbed something. If we use ~inclusion to find distinguishable entities, then the set of properties of the entity which is a robber is ~included in the set of properties of the other, and so the entity which is a robber is not distinguishable. But we know that in an example like 'A robber meets a child who robs something. The robber ...', the definite description 'the robber' rather designates the first entity introduced as 'a roober' than the second one. And this implies that the first entity is distinguishable. So, we must *not* use the ~inclusion to distinguish entities (but only the simple inclusion).

The remaining problem is then that, in this case, the definite description 'the robber' is ~included in two distinguishable entities ('a robber' and 'a child who robs something'), and then can't be a *dd*, in the sense introduced in § 2.1. To take into account an example like the one above and the effect of the ~inclusion, we must refine the definition of what can be a *dd*: a *dd* must designate (in the sense of the ~inclusion) only one entity of the context. But if there is more than one, but only one entity *e* such that $\mathcal{P}_{dd} \subseteq \mathcal{P}_e$ (with the simple inclusion), then the *dd* is correct and the entity designated is *e*.

As shown below, we have used this inclusion while dealing with incomplete distinguishing description.

2.3 Algorithm for Idd production using simple inclusion or ~inclusion

Idd as defined in 2.1 do not depend on how they are produced. Here we give an algorithm which builds Idd from left to right, word by word, in order to be used in a guided composition system.

First of all, it can be seen that if a given *Idd* is not correct, any *Idd* starting with this *Idd* cannot be correct. Hence, we just need to examine one word strings first, then two word strings, and so on, which fits perfectly to the guided composition mode that we used in our applications (see §3).

So, the only words that the system must propose are those which continue a correct *Idd* into a correct *Idd*. At the beginning, the word 'le' ('the') can be proposed if and only if $\mathcal{D}_\mathcal{E}$ is not empty ($\mathcal{P}_{\text{'le'}} = \emptyset$). Thus, we have a set of distinguishable entities. A word w_1 can be proposed if and only if there exists at least one entity in this set that agrees with 'le w_1'. And so on.

The implemented algorithm works as follow:

Let us consider $w_1..w_n$ an *Idd*. At each step, two sets are built: one corresponding to all the entities that agree with $w_1..w_n$ (we call it $\mathcal{PR}_{w_1..w_n}$, the set of the possible referents of the *Idd*), one corresponding to the distinguishable entities that agree with it ($\mathcal{DPR}_{w_1..w_n}$, the distinguishable possible referents of $w_1..w_n$). If $\mathcal{DPR}_{w_1..w_n}$ is not empty, the *Idd* is correct and can be continued. If $\mathcal{PR}_{w_1..w_n}$ is a singleton (and if the *Idd* is an NP syntactically correct) then the *Idd* can be considered as a *dd*, and its referent is the element of the set. If $\mathcal{PR}_{w_1..w_n}$ is not a singleton, the system must propose words to continue the NP (which is always possible). The words w which are contextually correct are those for which the set $\mathcal{DPR}_{w_1..w_nw}$ remains not empty.

As seen above, the set $\mathcal{DPR}_{w_1..w_n}$ is built using simple inclusion.

The set $\mathcal{PR}_{w_1..w_n}$ can be constructed according to simple inclusion or ~inclusion. In the second case, to test if an *Idd* $w_1..w_n$ is a *dd*, the algorithm has to test if $\mathcal{PR}_{w_1..w_n}$ is a singleton or if a subset of it according to simple inclusion is a singleton.

We have the following properties:

$\mathcal{PR}$'the' = $\mathcal{E}$ $\mathcal{DPR}$'le' = $\mathcal{D}_{\mathcal{E}}$

For each correct *Idd* $w_1..w_n$:

$\mathcal{DPR}_{w_1..w_n} \subseteq \mathcal{PR}_{w_1..w_n}$

and for each word w:

$\mathcal{PR}_{w_1..w_n w} \subseteq \mathcal{PR}_{w_1..w_n}$

$\mathcal{DPR}_{w_1..w_n w} = \mathcal{PR}_{w_1..w_n w} \cap \mathcal{DPR}_{w_1..w_n}$

and so,

$\mathcal{DPR}_{w_1..w_n w} \subseteq \mathcal{DPR}_{w_1..w_n}$

These are fine properties which ensure that the algorithm stops. Computing $\mathcal{PR}_{w_1..w_n}$ and $\mathcal{DPR}_{w_1..w_n}$ is achieved easily for n>1: $\mathcal{PR}_{w_1..w_n}$ is composed of all those elements of $\mathcal{PR}_{w_1..w_{n-1}}$ that also agree with $w_1..w_n$. The same applies to $\mathcal{DPR}_{w_1..w_n}$. The only somewhat time-consuming task relies in computing $\mathcal{DPR}$'le'.

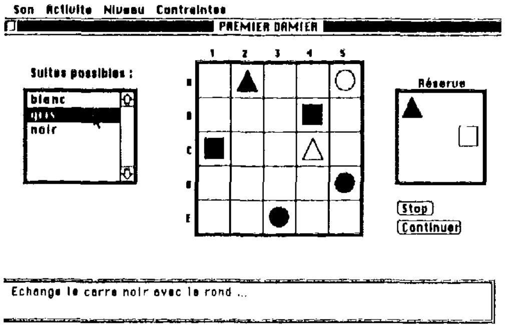
3.An application

The questions tackled in this paper have appeared while developping the system EREL[1] ([Godbert 98]) built on ILLICO. The generic ILLICO software ([Milhaud 92]) belongs to the category of guided composition systems we have presented above. It has been used to develop several natural language interfaces (see for example [Guenthner & al. 92], [Pasero & al. 94]). In this system, if a string proposed by the user is incorrect, then the system keeps the largest correct sub-string (from the beginning of the string) and proposes all the possible words that can continue this sub-string. If the sentence is empty, then the system can propose all the possible first words, and so on. It is important to notice that the generative process is driven by the syntactic parser.

EREL is a software for language rehabilitation that we have designed in a collaboration with medical staffs specialised in the treatment of autistic-like children. The system provides a set of user-friendly educational play activities designed to help users to employ common language.

So, it is very important in this kind of applications to offer a sophisticated guided composition mode, and in particular to produce only correct sentences at every level (syntactical, conceptual but also contextual). At the moment, there are two main types of games in EREL: in the first one, chidren speak about or ask questions about a picture they see on the screen. The context which contains objects about which the chidren can talk about is preliminary computed and it does not change along with the discourse. So, the set of distinguishable entities $\mathcal{D}_{\mathcal{E}}$ is built at one time.

The second activity concerns a dialog on a logic game in which users compose orders in natural language to achieve a goal. One of the exercises consists in putting and moving objects on a board. A child has an initial stock of objects that he can put on a checker board, permute, move, or stow away. He gives orders to the system using natural languages sentences and he can see immediately on the board the effects the sentences have. The interface looks like this:



Echange le carre noir avec le rond ...

Here, in the French version, the user has begun a sentence «*Echange le carré noir avec le rond...*» (*Permute the black square with the circle...*) and the system, according to the contextual situation, proposes the possible words to be selected: *blanc, gris, noir* (*white, grey, black*). If there had been no white circle on the board, the word *white* would not have been proposed.

We have also implemented part of the ~inclusion presented above so the child can use the hyponym *pawn* to designate a triangle or a square or a circle. The hyponym relations are already represented in the

[1] The project EREL is partially funded by the Conseil Général des Bouches-du-Rhône.

system by a conceptual graph which is used in other parts of the system.

In this game, it is clear that contextually correct definite descriptions must designate objects in the context (the system has to act on them, so it has to find them). So, in the guided composition mode, the system has to compute *Idd* as they have been defined in §2. The objects are moving during the game so, as opposed to the first type of activity, the context changes and the set of distinguishable entities has to be computed at every step. Moreover, the children can create new pawns (made from a predefined set of shapes and colors). These objects are added in the context and must be taken into account for later mentions.

The set of distinguishable entities is not the set of all the objects because some objects may not be distinguishable. For example, if two objects have the same shape and color and are in the stock ('Réserve' on the figure above), then they can not be distinguished (we assume there is no relative position in the reserve). The user can act on them by using sentences like 'put one of the red triangles which are in the reserve in the case A4' but not like 'put the red triangle which is in the reserve...'. If there is no other triangle on the chessboard, then the system must not propose beginnings of definite NP like 'the triangle...'

EREL is under development and a medical team who works with autistic children is testing a preliminary version.

4.Related works

The work presented here uses notions firstly introduced by [Dale 89] and mentioned in many works in the field. We have presented here new applications and extensions. Actually, the problem treated here raises the general question of generating definite descriptions. Generally, these works deal rather with the problem of *what to say* and *how to say it.* [Novak 88] deals with the problem (among others) of when and how restrictives relative clause have to be used in definite NP. The system that he describes is able to produced definite NP like *the first yellow BMW*, *the second yellow BMW* if necessary. [Kronfeld 89] talks about 'conversionally relevant descriptions' which is typically the problem of *how to say* something according to the context of discourse or the user's goal like in [Appelt 85]. The relations between Gricean maxims and the generation of definite NP are studied in [Passonneau 95] and [Dale & al. 96] for example.

[Horacek 97] gives a good comparison of the previous works; his analyses make appear the problem of the linguistic realisation of a set of properties of an entity to generate a description that designates it and he proposes an algorithm which takes into account this problem during the choice of the property which will be used to build a *dd.*

Concerning the production of *Idd,* we are not really confronted with the problems mentioned in [Horacek 97] because the guided composition system doesn't generate NPs from entity representation; its parser generates partial syntactically correct sentences which are filtered by contextual criteria (the processes are driven in parallel thanks to coroutined methods). Moreover, concerning what to say and how to say it, it is the user who chooses what word (among the possibilities offered by the system) will be kept to build the sentence, at every step.

Concerning the extension of the notion of agreement that we make (and so of the notion of distinguishing description), many linguists mention the phenomenon we want to take into account. A more computational point of view is discussed in [Groenendijk & al 96, pp 25-27] (the example of 'the doctor' and 'the man'). The authors do not give really computable solutions to this problem. It seems that the use of simple inclusion and ~inclusion to find distinguishable entity and to identify a referent for a (complete or incomplete) distinguishing description (as described in §2.2) deals rather efficiently with the problem

5.Conclusion

We showed here how the uniqueness requirement, when dealing with incomplete definite descriptions, turns into a requirement of that particular sort of entities from the context, the distinguishable entities. Then we showed how the notion of distinguishing description can be extended using inclusion and what we called ~inclusion. An algorithm that uses these ideas and allows to know as early as possible incomplete definite description that can lead to correct definite description from those that cannot is given. The algorithm is incremental, which is particulary useful in a guided composition system and allows also to solve complete definite description (finding the referent). So far, an instance of it has been implemented under the system EREL.

Bibliography

[Appelt 85] Douglas E. Appelt. "Planning English Referring Expressions", *Artificial Intelligence*, Vol. 26, n° 1, April 1985.

[Corblin 95] Francis Corblin. *Les formes de reprises dans le discours. Anaphores et chaînes de référence,* Presses Universitaires de Rennes, 1995.

[Dale 89] Robert Dale. "Cooking Up Referring Expressions.", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver BC, 1989.

[Dale & al 96] Robert Dale and Ehud Reiter. "The Role of the Gricean Maxims in the Generation of Referring Expressions", *AAAI Spring Symposium on Computational Model of Conversational Implicature*, 1996.

[Danlos 85] Laurence Danlos. *Génération Automatique de Texte en Langage Naturel*, Masson, 1985.

[Godbert 98] Elisabeth Godbert. "EREL: a multimedia CALL system devoted to children with language disorders", In *Multimedia CALL : Theory and Practice*, K. Cameron, Ed. Elm Bank Publications, Exeter, England, 1998, pp. 207-216.

[Groenendijk & al 96] Jeroen Groenendijk and Martin Stokhof "Changez le Contexte!", Research report, ILLC/Département of Philosophy, University of Amsterdam, March 1996.

[Guenthner & al 92] Frantz Guenthner, Karin Kruger-Thielmann, Robert Pasero and Paul Sabatier, "Communications Aids for ALS Patients", *Proceedings of the 3rd International Conference on Computers for Handicapped Persons*, pp. 303-307, 1992.

[Horacek 97] Helmut Horacek, "An Algorithm For Generating Referential Descriptions With Flexible Interfaces", *Proceedings of the 35th Annual Meeting of ACL and 8th Annual Meeting of EACL*, Madrid, Spain, 1997.

[Kronfeld 89] Amichai Kronfeld "Conversationally Relevant Descriptions", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver BC, 1989.

[Milhaud 92] Gerard Milhaud, Robert Pasero and Paul Sabatier "Partial Synthesis of Sentences by Coroutining Constraints on Different Levels of Well-Formedness", *Proceedings of the 15th International Conference on Computational Linguistics* (Coling 92), pp 926-929, Nantes, France, 1992.

[Novak 88] Hans-Joachim Novak "Generating Referring Phrases in a Dynamic Environment", Chapter 5 in M. Zock and G. Sabah (eds), *Advances in Natural Language Generation*, Volume 2, pp76-85, Pinter Publishers, 1988.

[Pasero 94] Robert Pasero, Nathalie Richardet and Paul Sabatier, "Guided Sentences Composition for Disabled People", *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pp.205-206, 1994.

[Passonneau 95] Rebecca J. Passonneau, "Integrating Gricean and Attentional Constraints", *Proceedings of the International Joint Conference on Artificial Intelligence*, Montréal, Quebec, 1995.

[Rincel & al 89] Paul Rincel and Paul Sabatier, "LEADER : Un generateur d'interfaces en langage naturel pour bases de données relationnelles", *Proceedings of the AFCET RFIA Conference*, Paris, 1989.