Degree of Abstraction in Referring Expression Generation and its Relation with the Construction of the Contrast Set

Raquel Hervás
Facultad de Informática
Universidad Complutense de Madrid
Madrid, Spain
raquelhb@fdi.ucm.es

Pablo Gervás
Facultad de Informática
Universidad Complutense de Madrid
Madrid, Spain
pgervas@sip.ucm.es

Abstract

Referring Expression Generation (REG) is the task that deals with references to entities appearing in a spoken or written discourse. If these referents are organized in terms of a taxonomy, there are two problems when establishing a reference that would distinguish an intended referent from its possible distractors. The first one is the choice of the set of possible distractrors or *contrast set* in the given situation. The second is to identify at what level of the taxonomy to phrase the reference so that it unambiguously picks out only the intended referent, leaving all possible distractors in different branches of the taxonomy. We discuss the use of ontologies to deal with the REG task, paying special attention to the choice of the the contrast set and to the use of the information of the ontology to select the most appropriate type to be used for the referent.

1 Introduction

Referring Expression Generation (REG) is the task that deals with references of entities appearing in a discourse. In a context where possible referents are organized in terms of a taxonomy (or subsumption hierarchy) and may additionally be differentiated by their attributes, there are two possible ways of establishing a reference that will distinguish an intended referent from its possible distractors.

One is to identify at what level of the taxonomy to phrase the reference so that it unambiguously picks out only the intended referent, leaving all possible distractors in different branches of the taxonomy. Another, applied once a particular level of reference has been chosen, is to resort to mentioning additional attributes of the intended referents that distinguish it from any remaining distractors that share the same branch of the taxonomy.

While the second task has been addressed often in existing literature, the first one is often glossed over by requiring that the levels to be used for each element come specified in the input. However, if this task is to be considered as a specific problem to be solved computationally, it opens up an additional problem. If the elements in the universe are classified in a taxonomy with a single root and the reference was established at a high enough level in the taxonomy, potentially everything in the universe could be a distractor for any other element.

In this paper we will discuss the use of ontologies to deal with the referring expression generation task. We will pay special attention to the choice of the contrast set and the use of ontology information to select the most appropriate type to be used for the referent. This work has been centered in the generation of definite noun phrases where the type of an element and a set of its properties are given to distinguish it from the other elements in focus. We are also supposing that the situations in which the reference is produced are static, that is, the addressee's perception of the world does not change during the process of reference generation.

2 Related Work

The appropriate use of referring expressions to compete with human-generated texts involves a certain difficulty. According to Reiter and Dale (2000), a referring expression must communicate enough infor-

mation to identify univocally the intended referent within the context of the current discourse, but always avoiding unnecessary or redundant modifiers.

Reiter and Dale (1992) describe a fast algorithm for generating referring expressions in the context of a natural language generation system. Their algorithm relies on the following set of assumptions about the underlying knowledge base that must be used: (1) every entity is characterized in terms of a collection of attributes and their values, (2) every entity has as one of its attributes a type, and (3) the knowledge base may organize some attribute values as a subsumption hierarchy. Additionally, each object represented in the system should have an associated *basic level value*, which corresponds to the concept which is preferred when referring to that object.

These assumptions are satisfied if a description logic ontology is used for this purpose. Entities would correspond to instances of concepts from the ontology, the attribute corresponding to the type would be the concept of which they are immediate instances, and the taxonomical structure of the ontology of concepts would provide the subsumption hierarchy. To construct a reference to a particular entity, the algorithm takes as input a symbol corresponding to the intended referent and a list of symbols corresponding to other entities in focus, known as the *contrast set*. The algorithm returns a list of attribute-value pairs that correspond to the semantic content of the referring expression.

3 Generating References Using Ontologies

A previously developed ontology about wines has been used to test the ideas presented in this work. This is a sample ontology implemented following a version published by Brachman and colleagues (Brachman et al., 1991) and distributed along with the CLASSIC knowledge representation system.

We have focused on the taxonomy of wines provided by the ontology. Wines are divided in three main categories: `Red_Wine`, `White_Wine` and `Rose_Wine`. Inside these main categories there is a complex taxonomy of different kinds of wines. In addition, the ontology also contains several instances of the different concepts. Each of these instances is described using features such as body, color, flavor, producer, and so on.

The aim is to generate references for different instances of wines which are together in a discourse. The first step is to select the set of distractors or contrast set for the specific referent. Then, an algorithm for deciding which is the best reference to use is applied. We have considered as the best reference possible the use of the type that distinguishes the referent from the distractors and at the same time is as general as possible. For example, if we are referring to an instance of `Chardonnay` wine (that is a white one) in a situation where the rest of wines are all red wines, the most suitable reference is *"the white wine"* and not *"the chardonnay"*. On the contrary, the more specific (but unnecessary) reference might lead the addressee to infer that this information is somehow relevant. If only white wines (as direct type of the referent) are considered for the contrast set, only the more specific (and inappropriate) reference may be generated. Therefore, a wide enough contrast set must be considered in each case.

Finally, if the type chosen is not enough to distinguish the referent from the contrast set, attribute selection is applied to select a subset of the element properties that distinguish it.

3.1 Composing the Contrast Set

Information about type is generally used to determine which elements of the world must be considered in the contrast set. In this work, all the information about the world is located in an ontology. Each instance of the world contained in it has a direct type (the most specific concept it belongs to) and a set of undirect types that are all the types between the direct type and the root of the ontology.

In the work developed we have used the whole ontology as contrast set. We have considered it as the most suitable option for most situations where the elements involved can belong to quite different types. As we will see later, this choice avoids the use of references more specific than desired while at the same time it allows the algorithm to choose the type that is more suitable in a given situation.

3.2 An Appropriate Type for the Referent

Our approach takes as initial distinguishing attribute the type of the elements appearing in the world. This kind of solution is enough when the types defined for each of the entities of the world are fixed and there is

not a close relation for different types. For example, a solution that takes as type the strict one defined in an element would not consider a *doberman* and a *chihuahua* as being both of them *dogs*.

The algorithm we have implemented can be seen in Figure 1. Here, *r* is the intended referent, *C* is the contrast set, *A* is the list of attributes that the instances of the ontology hold, *typeValue* is the type that would be assigned to the referent by the algorithm, and *L* is the list of attribute-value pairs returned if the type is not enough to rule out all the distractors. The `rules-out` function works as the one used in the Incremental algorithm, and the function `incremental-algorithm` calls directly to the original algorithm by Reiter and Dale.

The function `find-best-value-type` is the one that delivers the most appropriate type for the intended referent *r* taking into account the information in the ontology. We have considered as basic level value for the type the most specific of the common types of the instances of the ontology. From this basic level type, the branch of concepts between it and the direct type of the intended referent *r* is visited. The type that will be used in the reference is the most general concept from this branch that discards a bigger number of distractors.

make-referring-expression (r, C , A)

$L \leftarrow \{\}$
$C \leftarrow$ instances-ontology()
$typeValue \leftarrow$ find-best-value-type(r, C)
$C \leftarrow C -$ rules-out($<$ type , $typeValue >$)
if $C = \{\}$ **then**
  return $< typeValue, \{\} >$
else
  $L \leftarrow$ incremental-algorithm(r , C , A)
  return $< typeValue, L >$
endif

find-best-value-type(r, C)

$basic\text{-}level\text{-}type \leftarrow$ most-specific-common-type($r \cup C$)
$value \leftarrow basic\text{-}level\text{-}type$
for $v_i \in$ ontology-children($basic\text{-}level\text{-}type$)
  if | rules-out($<$ type,$v_i >$) |
    $>$ | rules-out($<$ type,$value >$) | **then**
    $value \leftarrow v_i$
  endif
next
return $value$

Figure 1: The Algorithm

3.3 Attribute Selection for Reference Completion

In some cases the type would not be enough to distinguish a referent from the other elements of the world. This situation is produced when they belong to the same type. In this situation it will be necessary to use their properties to distinguish between them. The attribute selection carried out in the Incremental algorithm from Reiter and Dale has been applied to these situations.

4 Some Examples

We have tested the implemented algorithm over different situations in which a set of wines is presented. For each of them, a distinguishing description is provided using the appropriate type found using the ontology and a set of attributes when they were required. The instances of the world we have considered are shown in Table 1 (the properties of the wines that have not been used by the algorithm are not shown). The references generated for each of the referents are (numbers correspond to examples in the table):

1. *"The Riesling"*. There is another white wine but not a `Dry_Riesling` one, so the most general type discarding all the distractors is `Riesling`.

2. *"The moderate Cabernet_Sauvignon"*. Here the type is not enough to distinguish this referent, so its attributes are used. The property that distinguish it from the other `Cabernet_Sauvignon` is the flavor.

3. *"The strong Cabernet_Sauvignon"*. As in the previous case the strong flavor is used to distinguish the wine from the other `Cabernet_Sauvignon`.

4. *"The Rose_Wine"*. In this case there are no more rose wines, so this generic type is enough to distinguish the referent.

	Name of the instance	Types	Properties			
			Body	Color	Flavor	Sugar
1	Corban_Dry_White_Riesling	Wine White_wine Riesling Dry_Riesling Corbans_Dry_White_Riesling Chardonnay Bancroft_Chardonnay Rose_wine White_Merlot Forest_Glen_White_Merlot_Rose Red_wine Cabernet_Sauvignon Marietta_Cabernet_Sauvignon Forman_Cabernet_Sauvignon				
2	Marietta_Cabernet_Sauvignon		Medium	Red	Moderate	Dry
3	Forman_Cabernet_Sauvignon		Medium	Red	Strong	Dry
4	Forest_Glen_White_Merlot_Rose					

Table 1: Examples

5 Conclusions and Future Work

The main advantage of this approach is that the algorithm always finds the most suitable value for the type, taking into account the other entities of the world. Since this solution is completely generic and domain-independent, the algorithm would work in the same way with more general ontologies. For example, if the considered ontology contains information not only about wines, but also about other kinds of drinks, the values to be used as types of the referents would also be chosen in the same way. In this situation the referent could be the only wine among other drinks, and the reference generated for it would be the most appropriate one: *"the wine"*.

In the Incremental algorithm, Reiter and Dale do not address the question of how the contrast set is constructed, stating that the contrast set is one of the inputs of their algorithm. In our work, we have chosen as contrast set all the instances that can be found in the ontology. This solution allows the algorithm to work with enough information to choose exactly at which level of the ontology the discourse is being displayed (more general or more specific). With this information the generated references are adapted to the level of specificity required in each case.

The Incremental algorithm also states that the basic level value is obtained from the knowledge base or the user model. In this paper we have implemented a dynamic way to obtain this value that only depends on the knowledge available about the world. However, the use of some kind of user model representing the expertise level of the addressee in a specific domain could be explored in the future.

Acknowledgments

This research is funded by the Spanish Ministry of Education and Science (TIN2006-14433-C02-01 project) and the UCM and the Dirección General de Universidades e Investigación of the CAM (CCG07-UCM/TIC-2803).

References

Brachman. Ronald J. and McGuiness, Deborah L. and Patel-Schneider, Peter F. and Resnick, Lori A. 1991. *Living with CLASSIC: when and how to use a KL-ONE-like language.* Principles in Semantic Networks: Explorations in the Representation of Knowledge, pages 401–456. Morgan Kaufmann, California.

Reiter, Ehud and Dale, Robert. 1992. A fast algorithm for the generation of referring expressions. *Proc. of the 14th conference on Computational Linguistics*, pp. 232-238. Association for Computational Linguistics.

Reiter, Ehud and Dale, Robert. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.