

Generating Referential Descriptions Involving Relations by a Best-First Searching Procedure – A System Demo

Florin Haque

Saarland University
P.O. Box 151150
66041 Saarbrücken, GERMANY
s9flhaque@stud.uni-saarland.de

Helmut Horacek

German Research Center for AI
Stuhlsatzenhausweg 3
66123 Saarbrücken, GERMANY
helmut.horacek@dfki.de

Abstract

Despite considerable research invested in the generation of referring expressions (GRE), there still exists no adequate generic procedure for GRE involving relations. In this paper, we present a system for GRE that combines attributes and relations, using best-first search technique. Preliminary evaluations show its effectiveness; the design enables the use of heuristics that meet linguistic preferences.

1 Motivation

Empirical evidence shows that humans use relations in GRE more often than necessary [Viethen, Dale 2008]. Nevertheless, algorithms involving relations, starting with [Dale, Haddock 1991] still have not reached a significant level of rigour and coverage (the method by [Krahmer, van Erk, Verleg 2003] does, to a certain extent). In particular, the incremental algorithm [Dale, Reiter 1995] constitutes a severe commitment for GRE involving relations, because the choice among alternate referents related to the intended one leads to substantial differences at early phases.

In order to remedy this problem, we have applied best-first searching (A^*) to the issue at hand, as already explored for references to sets of objects involving boolean combinations of attributes [Horacek 2003]. This method yields the expression considered best according to the evaluation function used, with a guarantee of optimality, provided an *admissible* heuristic is built on the basis of the evaluation function.

2 General Approach and Some Specificities

Our approach applies the best-first search paradigm (as in [Horacek 2003]) to the conceptual algorithm described in [Horacek 1996], so that known unwanted effects (endless loops, unnecessary identification of objects) are avoided. Motivations, conceptualization and details of the implementation are described in [Haque 2015].

When searching for components of an adequate referring expression, a tree consisting of partial expressions describing the intended referent, also in terms of the objects related to it, is successively built. Tree expansion is geared by the A^* -specific function f , which is composed of the cost of a partial expression built so far (g) and the most optimistic estimate of reaching a goal state (h), i.e., in a single step. This process terminates once an identifying and provably best description has been found. It is speeded up by A^* specific and local similarity-based cut-offs.

The sum of g and h reflects the relative quality of competing partial descriptions. To impose a more fine-grained ordering over the candidates for the next descriptor to be tried out, we have used discriminatory power to resolve the ties.

1. Attributes and relations are treated in a uniform way. Relations are tried out after attributes by assigning lower costs to attributes, as relations require a description of the object related (attributes may suffice alone).
2. A relation may be chosen even if it applies to all potential distractors, but only if all the objects possessing this relation are not related with the same object via this relation.

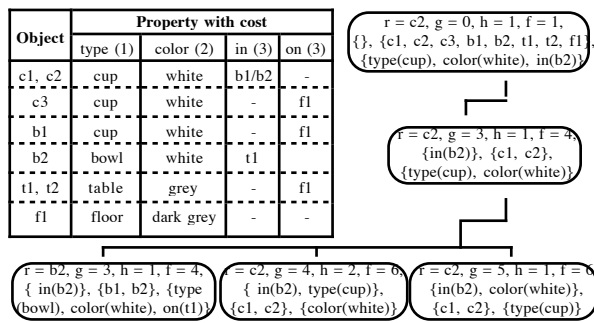


Figure 1. A small scenario and a fragment of the search tree

3 Implementation

The algorithm is implemented in C++, running on an Intel Core i5 processor with 1.6 GHz.

The functions g and h can be parameterized context-independently. For the test scenarios, we have used simple counts for each part, such as 1 for type, 2 for other attributes, and 3 for relations, so that the shortest expression results.

At first, we have tested the system with a few scenarios similar to those discussed in the literature – a room with tables, bowls, cups, etc., with some attributes (e.g., *type*, *color*) and relations (e.g., spatial containment – ‘*in*’, spatial support – ‘*on*’, left-of, and right-of). Figure 1 (top left) shows such a scenario (*cup c2* being the intended referent), and a portion of the search tree that illustrates the expansion of a node via a relation (in the node structure ‘ r ’ is the local referent, and the last three sets include accumulated descriptors, context set, and available properties, respectively). It finally leads to the identifying expression “*the cup in the bowl on the table*”. To check how the system handles relatively complex situations, we have designed a scenario composed of 40 entities with 10 well-defined descriptors (4 attributes and 6 relations).

Table 1 summarizes the results for some small scenarios (2nd line for the scenario from [Horacek 1996] and 3rd line for the scenario from [Dale, Haddock 1991]) and for the extended one (last line), in terms of tree size and running time (ranging from smallest to largest). For the extended scenario, easy identification tasks do not require extra resources in comparison to the small scenarios. In contrast, identification of a specific bottle needed the largest tree (269 no-

des) and longest run-time (298 msec) incorporating four chained relations in the generated expression which can be glossed as ‘*the bottle in the bowl which is in a plate on the table under which there is a glass*’.

The system is always able to find a reasonable expression without extra components, some including several attributes and relations. Since the evaluation functions used so far do not express subtle preferences, several ties may result. For example, “the metal bottle on the table”, “the metal bottle right of a glass”, “the white bottle right of a glass”, “the bottle right of a glass with water” are produced as equivalent alternatives for identifying one specific bottle in the extended scenario.

4 Conclusion and Extensions

In this paper, we have presented an approach for generating referential descriptions involving relations by a best-first searching procedure. The system is able to find the best expression (or multiple equally good expressions if exist) according to the evaluation function used. For the examples we have tested so far, the resulting expressions are reasonable and the computation times needed are very convincing.

In further developing the system, we envision conceptual extensions, such as the use of negation (“the table on which there are no bottles”, “the empty table”). Moreover, we need to make technical refinements, most importantly the use of context-sensitive evaluation functions for the resulting expressions, especially to cater for situation-dependent uses of descriptors redundant for identification purposes; the challenge here is to derive heuristic functions that are still admissible. In addition, we intend to test the system in larger and more diverse situations, preferably backed-up by corpus data.

<i>no. of entities</i>	<i>no. of tree nodes</i>	<i>time (msec)</i>
4	4 to 7	1 to 4
6	4 to 24	1 to 18
8	4 to 7	1 to 5
40	4 to 269	2 to 298

Table 1. Summary of searches for a few small scenarios and an extended one

References

- Dale, R., and Haddock, N. 1991. Generating Referring Expressions Involving Relations. Proceedings of the *27th Annual Meeting of the European Chapter of the ACL (EACL'91)*, pp. 161-166, Berlin, Germany.
- Dale, R., and Reiter, E. 1995. Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science* 18, pp. 233-263.
- Haque, F. 2015. Generating Referring Expressions Involving Relations by Best-First Searching. Master thesis, Saarland University, Department of Computer Science.
- Horacek, H. 1996. A New Algorithm for Generating Referential Descriptions. In Proc. of *12th European Conference on Artificial Intelligence (ECAI-96)*, pp. 577-581, Budapest, Hungary.
- Horacek, H. 2003. A Best-First Search Algorithm for Generating Referring Expressions. In Proc. of the *European Chapter of the ACL (EACL'2003)*, pp. 206-213, Budapest, Hungary.
- Krahmer, E., v. Erk S., and Verleg, A. 2003. Graph-based Generation of Referring Expressions. *Computational Linguistics*, 29(1), pp. 53-72.
- Viethen, J. and Dale, R. 2008. The Use of Spatial Relations in Referring Expressions. In Proceedings of the *5th International Natural Language Generation Conference (INLG)*, pp. 59-67, Salt Fork, OH.