GRAPH: The Costs of Redundancy in Referring Expressions

Emiel Krahmer
Tilburg University
The Netherlands
e.j.krahmer@uvt.nl

Mariët Theune
University of Twente
The Netherlands
m.theune@utwente.nl

Jette Viethen
Macquarie University
Australia
jviethen@ics.mq.edu.au

Iris Hendrickx
University of Antwerp
Belgium
iris.hendrickx@ua.ac.be

Abstract

We describe a graph-based generation system that participated in the TUNA attribute selection and realisation task of the REG 2008 Challenge. Using a stochastic cost function (with certain properties for free), and trying attributes from cheapest to more expensive, the system achieves overall .76 DICE and .54 MASI scores for attribute selection on the development set. For realisation, it turns out that in some cases higher attribute selection accuracy leads to larger differences between system-generated and human descriptions.

1 Introduction

Referring Expression Generation (REG) is a key-task in NLG, and the topic of the REG 2008 Challenge.[1] In this context, referring expressions are understood as *distinguishing descriptions*: descriptions that uniquely characterize a target object in a visual scene (e.g., "the red sofa"), and do not apply to any of the other objects in the scene (the distractors). Generating such descriptions is usually assumed to be a two-step procedure: first, it has to be decided which attributes of the target suffice to characterize it uniquely, and then the selected set of attributes should be converted into natural language.

For the first step, attribute selection, we use a version of the Graph-based REG algorithm of Krahmer et al. (2003). In this approach, a visual scene is represented as a directed labelled graph, where vertices represent the objects in the scene and edges their attributes. A key ingredient of the approach is that costs can be assigned to attributes; the generation of referring expressions can then be defined as a graph search problem, which outputs the cheapest distinguishing graph (if one exists) given a particular cost function. For the second step, realisation, we use a simple template-based realiser written by Irene Langkilde-Geary from Brighton University that was made available to all REG 2008 participants.

A version of the Graph-based algorithm was submitted for the ASGRE 2007 Challenge (Theune et al. 2007). For us, one of the most striking, general outcomes was the observed "trend for the mean DICE score obtained by a system to decrease as the proportion of minimal descriptions increases" (Belz and Gatt 2007).[2] Thus, while REG systems have a tendency to produce minimal descriptions, human speakers tend to include redundant properties in their descriptions, which is in line with recent findings in psycholinguistics on the production of referring expressions (e.g., Engelhardt et al. 2006).

In principle, the graph-based approach has the potential to deal with redundancy by allowing some attributes to have zero costs. Viethen et al. (2008), however, show that merely assigning zero costs to an attribute is not a sufficient condition for inclusion; if the search terminates before the free properties are tried, they will not be included. In other words: the order in which attributes are tried should be explicitly controlled as well. In the experiment we describe here, we consider both these factors and their interplay.

[1] See http://www.itri.brighton.ac.uk/research/reg08/.

[2] DICE (like MASI) is a measure for similarity between a predicted attribute set and a (human produced) reference set.

2 Method

We experimentally combine four cost functions and two search orders (Table 1). (1) **Simple** simply assigns each edge a 1-point cost. (2) **Stochastic** associates each edge with a frequency-based cost, based on both the 2008 training and development sets (assuming that a larger data set allows for more accurate frequency estimates). (3) **Free-Stochastic** is like the previous cost function, except that highly frequent attributes are assigned 0 costs. For the Furniture domain, this applies to “colour”; for People to “hasBeard = 1” and “hasGlasses = 1.” (4) **Free-Naive**, finally, reduces the relatively fine-grained costs of Free-Stochastic to three values (0 = free, 1 = cheap, 2 = expensive). In addition, we compare results for two property orderings: (A) Properties are tried in a **Random** order. (B) **Cost-based**, where properties are tried (in stochastic order) from cheapest to most expensive. Finally, since human speakers nearly always include the “type” property, we decided to simply always include it. Tables 2 to 4 summarize the evaluation results for all combinations of cost functions and search orders.

Table 1: Overview of cost functions and search orders. The GRAPH 4+B settings were submitted to the REG 2008 Challenge.

Costs		**Orders**	
1	Simple	A	Random
2	Stochastic	B	Cost-based
3	Free-stochastic		
4	Free-naive		

3 Attribute Selection Results

The measures used to evaluate attribute selection are DICE, MASI, attribute accuracy (A-A, the proportion of times the generated attribute set was identical to the reference set), and minimality (MIN).

Notice first that the order in which attributes are tried in the search process matters; the B-systems nearly always outperform their A-counterparts. Second, assigning varying costs also helps; both 1-variants (**Simple** costs) perform worse than the systems building on **Stochastic** cost functions (2, 3 and 4). Third, adding free properties is also beneficial; the 3 and 4 variants clearly outperform the 1 and 2 variants. It is interesting to observe that the **Free-naive** cost function (4) performs equally well as the more principled **Free-stochastic** (3), but only in combination with the **Cost-based** order (B). To the extent that it is possible to compare the results, the submitted GRAPH 4+B outperforms our best 2007 variant (GRAPH FP in Table 2). This suggests that the interplay between property ordering and cost function is a flexible and efficient approach to attribute selection.

Table 2: Furniture development set results (80 trials).

GRAPH	DICE	MASI	A-A	MIN	EDIT	S-A
1+A	.61	.32	.12	.29	5.90	.04
1+B	.61	.31	.12	.29	5.89	.04
2+A	.71	.47	.31	.11	5.06	.05
2+B	.69	.44	.28	.16	5.19	.05
3+A	.80	.58	.45	.00	4.90	.05
3+B	.80	.58	.45	.00	4.90	.05
4+A	.80	.59	.48	.00	4.61	.05
4+B	.80	.59	.48	.00	4.61	.05
FP 2007	.71	–	–	–	–	–

Table 3: People development set results (68 trials).

GRAPH	DICE	MASI	A-A	MIN	EDIT	S-A
1+A	.59	.36	.24	.00	6.54	.00
1+B	.66	.42	.24	.00	6.78	.00
2+A	.66	.42	.24	.00	6.78	.00
2+B	.66	.42	.24	.00	6.78	.00
3+A	.68	.41	.19	.00	6.79	.00
3+B	.72	.48	.28	.00	6.96	.00
4+A	.59	.34	.18	.00	6.56	.00
4+B	.72	.48	.28	.00	6.96	.00
FP 2007	.67	–	–	–	–	–

Table 4: Combined Furniture and People development set results.

GRAPH	DICE	MASI	A-A	MIN	EDIT	S-A
1+A	.60	.34	.18	.16	6.20	.02
1+B	.63	.36	.18	.16	6.30	.02
2+A	.69	.45	.28	.06	5.85	.03
2+B	.68	.43	.26	.09	5.92	.03
3+A	.74	.51	.33	.00	5.77	.03
3+B	.76	.54	.37	.00	5.84	.03
4+A	.70	.48	.34	.00	5.51	.03
4+B	.76	.54	.39	.00	5.69	.03
FP 2007	.69	–	–	–	–	–

4 Realization Results

To evaluate realisation, the following two word-string comparison measures were used: string-edit distance (EDIT), which is the Levenshtein distance between generated word string and human reference output, and string accuracy (S-A), which is the proportion of times the word string was identical to the reference string.

For all settings of the algorithm, we see that S-A is much lower than A-A. This is as expected, since any set of attributes can be expressed in many different ways, and the chance that the realizer produces exactly the same string as the human reference is quite small. For the furniture domain, we see that S-A has a fairly constant low score, while EDIT follows the same pattern as A-A: including redundant (free) properties leads to better results. For the people domain, S-A is always 0, and surprisingly EDIT gets worse as A-A gets better.

To explain these results, we inspect those descriptions where A-A = 1 but S-A = 0, i.e., the attribute set is identical to the human reference but the word string is not. In setting 4+B (submitted to REG 2008) this is the case for 34 furniture and 19 people descriptions. For furniture, we see that the low S-A score can be largely explained by the fact that in 23 of the 34 descriptions the human reference either included no determiner or an indefinite one, whereas the system always included a definite determiner. This also explains why S-A hardly improves with higher A-A scores, since determiner choice is independent from attribute selection.

In the people domain, the zero scores for S-A can be explained by the fact that the realizer always uses "person" to express the type attribute, where the human references have either "man" or "guy" (in line with the human preference for *basic level* values; cf. Krahmer et al. 2003). We also encounter the determiner problem again, aggravated by the fact that many person descriptions include embedded noun phrases (e.g., "man with beard").

To find out why EDIT gets worse as A-A increases for different system settings in the people domain, we look at the six descriptions that have A-A = 1 for setting 4+B but not for 4+A. It turns out that five of these descriptions are realized as "the light-haired person with a beard", while the human reference strings are variations of "the man with a white beard", resulting in a relatively high EDIT value. The problem here is that the link between beard and hair colour has been lost in the data annotation process.

In general, we can conclude that simply combining more or less human-like attribute selection with an off-the-shelf surface realiser is not sufficient to produce human-like referring expressions.

Acknowledgements We thank the REG 2008 organizers for making the realiser available, and Hendri Hondorp for his help with installing and using it.

References

Belz, A. and A. Gatt 2007. The attribute selection for GRE challenge: Overview and evaluation results *Proceedings of UCNLG+MT* 75-83

Engelhardt, P., K. Bailey and F. Ferreira 2006. Do speakers and listeners observe the Gricean Maxim of Quantity? *Journal of Memory and Language, 54*, 554-573.

Krahmer, E., S. van Erk and A. Verleg 2003. Graph-based generation of referring expressions. *Computational Linguistics, 29(1)*, 5372.

Theune, M., P. Touset, J. Viethen, and E. Krahmer. 2007. Cost-based attribute selection for generating referring expressions (GRAPH-FP and GRAPH-SC). *Proceedings of the* ASGRE *Challenge 2007*, Copenhagen, Denmark

Viethen, J., R. Dale, E. Krahmer, M. Theune and P. Touset. 2008. Controlling redundancy in referring expressions. *Proceedings LREC 08*, Marrakech, Morroco.