Does Size Matter – How Much Data is Required to Train a REG Algorithm?

Mariët Theune
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands
m.theune@utwente.nl

Ruud Koolen
Tilburg University
P.O. Box 90135
5000 LE Tilburg
The Netherlands
r.m.f.koolen@uvt.nl

Emiel Krahmer
Tilburg University
P.O. Box 90135
5000 LE Tilburg
The Netherlands
e.j.krahmer@uvt.nl

Sander Wubben
Tilburg University
P.O. Box 90135
5000 LE Tilburg
The Netherlands
s.wubben@uvt.nl

Abstract

In this paper we investigate how much data is required to train an algorithm for attribute selection, a subtask of Referring Expressions Generation (REG). To enable comparison between different-sized training sets, a systematic training method was developed. The results show that depending on the complexity of the domain, training on 10 to 20 items may already lead to a good performance.

1 Introduction

There are many ways in which we can refer to objects and people in the real world. A chair, for example, can be referred to as red, large, or seen from the front, while men may be singled out in terms of their pogonotrophy (facial hairstyle), clothing and many other attributes. This poses a problem for algorithms that automatically generate referring expressions: how to determine which attributes to use?

One solution is to assume that some attributes are preferred over others, and this is indeed what many Referring Expressions Generation (REG) algorithms do. A classic example is the Incremental Algorithm (IA), which postulates the existence of a complete ranking of relevant attributes (Dale and Reiter, 1995). The IA essentially iterates through this list of preferred attributes, selecting an attribute for inclusion in a referring expression if it helps singling out the target from the other objects in the scene (the distractors). Crucially, Dale and Reiter do not specify how the ranking of attributes should be determined. They refer to psycholinguistic research suggesting that, in general, absolute attributes (such as color) are preferred over relative ones (such as size), but stress that constructing a preference order is essentially an empirical question, which will differ from one domain to another.

Many other REG algorithms similarly rely on preferences. The graph-based based REG algorithm (Krahmer et al., 2003), for example, models preferences in terms of costs, with cheaper properties being more preferred. Various ways to compute costs are possible; they can be defined, for instance, in terms of log probabilities, which makes frequently encountered properties cheap, and infrequent ones more expensive. Krahmer et al. (2008) argue that a less fine-grained cost function might generalize better, and propose to use frequency information to, somewhat ad hoc, define three costs: 0 (free), 1 (cheap) and 2 (expensive). This approach was shown to work well: the graph-based algorithm was the best performing system in the most recent REG Challenge (Gatt et al., 2009).

Many other attribute selection algorithms also rely on training data to determine preferences in one form or another (Fabbrizio et al., 2008; Gervás et al., 2008; Kelleher, 2007; Spanger et al., 2008; Viethen and Dale, 2010). Unfortunately, suitable data is hard to come by. It has been argued that determining which properties to include in a referring expression requires a "semantically transparent" corpus (van Deemter et al., 2006): a corpus that contains the actual properties of all domain objects as well as the properties that were selected for inclusion in a given reference to the target. Obviously, text corpora tend not to meet this requirement, which is why

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 660–664,
Portland, Oregon, June 19-24, 2011.

semantically transparent corpora are often collected using human participants who are asked to produce referring expressions for targets in controlled visual scenes for a given domain. Since this is a time consuming exercise, it will not be surprising that such corpora are thin on the ground (and are often only available for English). An important question therefore is how many human-produced references are needed to achieve a certain level of performance. Do we really need hundreds of instances, or can we already make informed decisions about preferences on a few or even one training instance?

In this paper, we address this question by systematically training the graph-based REG algorithm on a number of "semantically transparent" data sets of various sizes and evaluating on a held-out test set. The graph-based algorithm seems a good candidate for this exercise, in view of its performance in the REG challenges. For the sake of comparison, we also follow the evaluation methodology of the REG challenges, training and testing on two domains (a furniture and a people domain), and using two automatic metrics (Dice and accuracy) to measure human-likeness. One hurdle needs to be taken beforehand. Krahmer et al. (2008) manually assigned one of three costs to properties, loosely based on corpus frequencies. For our current evaluation experiments, this would hamper comparison across data sets, because it is difficult to do it in a manner that is both consistent and meaningful. Therefore we first experiment with a more systematic way of assigning a limited number of frequency-based costs to properties using k-means clustering.

2 Experiment I: k-means clustering costs

In this section we describe our experiment with k-means clustering to derive property costs from English and Dutch corpus data. For this experiment we looked at both English and Dutch, to make sure the chosen method does not only work well for English.

2.1 Materials

Our English training and test data were taken from the TUNA corpus (Gatt et al., 2007). This semantically transparent corpus contains referring expressions in two domains (furniture and people), collected in one of two conditions: in the -LOC condition, participants were discouraged from mentioning the location of the target in the visual scene, whereas in the +LOC condition they could mention any properties they wanted. The TUNA corpus was used for comparative evaluation in the REG Challenges (2007-2009). For training in our current experiment, we used the -LOC data from the training set of the REG Challenge 2009 (Gatt et al., 2009): 165 furniture descriptions and 136 people descriptions. For testing, we used the -LOC data from the TUNA 2009 development set: 38 furniture descriptions and 38 people descriptions.

Dutch data were taken from the D-TUNA corpus (Koolen and Krahmer, 2010). This corpus uses the same visual scenes and annotation scheme as the TUNA corpus, but with Dutch instead of English descriptions. D-TUNA does not include locations as object properties at all, hence our restriction to -LOC data for English (to make the Dutch and English data more comparable). As Dutch test data, we used 40 furniture items and 40 people items, randomly selected from the textual descriptions in the D-TUNA corpus. The remaining furniture and people descriptions (160 items each) were used for training.

2.2 Method

We first determined the frequency with which each property was mentioned in our training data, relative to the number of target objects with this property. Then we created different cost functions (mapping properties to costs) by means of k-means clustering, using the Weka toolkit. The k-means clustering algorithm assigns n points in a vector space to k clusters (S_1 to S_k) by assigning each point to the cluster with the nearest centroid. The total intra-cluster variance V is minimized by the function

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

where μ_i is the centroid of all the points $x_j \in S_i$. In our case, the points n are properties, the vector space is one-dimensional (frequency being the only dimension) and μ_i is the average frequency of the properties in S_i. The cluster-based costs are defined as follows:

$$\forall x_j \in S_i, cost(x_j) = i - 1$$

where S_1 is the cluster with the most frequent properties, S_2 is the cluster with the next most frequent properties, and so on. Using this approach, properties from cluster S_1 get cost 0 and thus can be added "for free" to a description. Free properties are always included, provided they help distinguish the target. This may lead to overspecified descriptions, mimicking the human tendency to mention redundant properties (Dale and Reiter, 1995).

We ran the clustering algorithm on our English and Dutch training data for up to six clusters ($k = 2$ to $k = 6$). Then we evaluated the performance of the resulting cost functions on the test data from the same language, using Dice (overlap between attribute sets) and Accuracy (perfect match between sets) as evaluation metrics. For comparison, we also evaluated the best scoring cost functions from Theune et al. (2010) on our test data. These "Free-Naïve" (FN) functions were created using the manual approach sketched in the introduction.

The order in which the graph-based algorithm tries to add attributes to a description is explicitly controlled to ensure that "free" distinguishing properties are included (Viethen et al., 2008). In our tests, we used an order of decreasing frequency; i.e., always examining more frequent properties first.[1]

2.3 Results

For the cluster-based cost functions, the best performance was achieved with $k = 2$, for both domains and both languages. Interestingly, this is the coarsest possible k-means function: with only two costs (0 and 1) it is even less fine-grained than the FN functions advocated by Krahmer et al. (2008). The results for the k-means costs with $k = 2$ and the FN costs of Theune et al. (2010) are shown in Table 1. No significant differences were found, which suggests that k-means clustering, with $k = 2$, can be used as a more systematic alternative for the manual assignment of frequency-based costs. We therefore applied this method in the next experiment.

3 Experiment II: varying training set size

To find out how much training data is required to achieve an acceptable attribute selection performance, in the second experiment we derived cost functions and property orders from different sized training sets, and evaluated them on our test data. For this experiment, we only used English data.

		Furniture		People	
Language	**Costs**	Dice	Acc.	Dice	Acc.
English	k-means	0.810	0.50	0.733	0.29
	FN	0.829	0.55	0.733	0.29
Dutch	k-means	0.929	0.68	0.812	0.33
	FN	0.929	0.68	0.812	0.33

Table 1: Results for k-means costs with $k = 2$ and the FN costs of Theune et al. (2010) on Dutch and English.

3.1 Materials

As training sets, we used randomly selected subsets of the full English training set from Experiment I, with set sizes of 1, 5, 10, 20 and 30 items. Because the accidental composition of a training set may strongly influence the results, we created 5 different sets of each size. The training sets were built up in a cumulative fashion: we started with five sets of size 1, then added 4 items to each of them to create five sets of size 5, etc. This resulted in five series of increasingly sized training sets. As test data, we used the same English test set as in Experiment I.

3.2 Method

We derived cost functions (using k-means clustering with $k = 2$) and orders from each of the training sets, following the method described in Section 2.2. In doing so, we had to deal with missing data: not all properties were present in all data sets.[2] For the cost functions, we simply assigned the highest cost (1) to the missing properties. For the order, we listed properties with the same frequency (0 for missing properties) in alphabetical order. This was done for the sake of comparability between training sets.

3.3 Results

To determine significance, we calculated the means of the scores of the five training sets for each set size, so that we could compare them with the scores of the entire set. We applied repeated measures of

[1] We used slightly different property orders than Theune et al. (2010), leading to minor differences in our FN results.

[2] This problem mostly affected the smaller training sets. By set size 10 only a few properties were missing, while by set size 20, all properties were present in all sets.

variance (ANOVA) to the Dice and Accuracy scores, using *set size* (1, 5, 10, 20, 30, entire set) as a within variable. The mean results for each training set size are shown in Table 2.[3] The general pattern is that the scores increase with the size of the training set, but the increase gets smaller as the set sizes become larger.

	Furniture		**People**	
Set size	Dice	Acc.	Dice	Acc.
1	0.693	0.25	0.560	0.13
5	0.756	0.34	0.620	0.15
10	0.777	0.40	0.686	0.20
20	0.788	0.41	0.719	0.25
30	0.782	0.41	0.718	0.27
Entire set	0.810	0.50	0.733	0.29

Table 2: Mean results for the different set sizes.

In the furniture domain, we found a main effect of *set size* (Dice: $F_{(5,185)} = 7.209$, $p < .001$; Accuracy: $F_{(5,185)} = 6.140$, $p < .001$). To see which set sizes performed significantly different as compared to the entire set, we conducted Tukey's HSD post hoc comparisons. For Dice, the scores of set size 10 ($p = .141$), set size 20 ($p = .353$), and set size 30 ($p = .197$) did not significantly differ from the scores of the entire set of 165 items. The Accuracy scores in the furniture domain show a slightly different pattern: the scores of the entire training set were still significantly higher than those of set size 30 ($p < .05$). This better performance when trained on the entire set may be caused by the fact that not all of the five training sets that were used for set sizes 1, 5, 10, 20 and 30 performed equally well.

In the people domain we also found a main effect of set size (Dice: $F_{(5,185)} = 21.359$, $p < .001$; Accuracy: $F_{(5,185)} = 8.074$, $p < .001$). Post hoc pairwise comparisons showed that the scores of set size 20 (Dice: $p = .416$; Accuracy: $p = .146$) and set size 30 (Dice: $p = .238$; Accuracy: $p = .324$) did not significantly differ from those of the full set of 136 items.

[3]For comparison: in the REG Challenge 2008, (which involved a different test set, but the same type of data), the best systems obtained overall Dice and accuracy scores of around 0.80 and 0.55 respectively (Gatt et al., 2008). These scores may well represent the performance ceiling for speaker and context independent algorithms on this task.

4 Discussion

Experiment II has shown that when using small data sets to train an attribute selection algorithm, results can be achieved that are not significantly different from those obtained using a much larger training set. Domain complexity appears to be a factor in how much training data is needed: using Dice as an evaluation metric, training sets of 10 sufficed in the simple furniture domain, while in the more complex people domain it took a set size of 20 to achieve results that do not significantly differ from those obtained using the full training set.

The accidental composition of the training sets may strongly influence the attribute selection performance. In the furniture domain, we found clear differences between the results of specific training sets, with "bad sets" pulling the overall performance down. This affected Accuracy but not Dice, possibly because the latter is a less strict metric.

Whether the encouraging results found for the graph-based algorithm generalize to other REG approaches is still an open question. We also need to investigate how the use of small training sets affects effectiveness and efficiency of target identification by human subjects; as shown by Belz and Gatt (2008), task-performance measures do not necessarily correlate with similarity measures such as Dice. Finally, it will be interesting to repeat Experiment II with Dutch data. The D-TUNA data are cleaner than the TUNA data (Theune et al., 2010), so the risk of "bad" training data will be smaller, which may lead to more consistent results across training sets.

5 Conclusion

Our experiment has shown that with 20 or less training instances, acceptable attribute selection results can be achieved; that is, results that do not significantly differ from those obtained using the entire training set. This is good news, because collecting such small amounts of training data should not take too much time and effort, making it relatively easy to do REG for new domains and languages.

Acknowledgments

Krahmer and Koolen received financial support from The Netherlands Organization for Scientific Research (Vici grant 27770007).

References

Anja Belz and Albert Gatt. 2008. Intrinsic vs. extrinsic evaluation measures for referring expression generation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 197–200.

Robert Dale and Ehud Reiter. 1995. Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Giuseppe Di Fabbrizio, Amanda Stent, and Srinivas Bangalore. 2008. Trainable speaker-based referring expression generation. In *Twelfth Conference on Computational Natural Language Learning (CoNLL-2008)*, pages 151–158.

Albert Gatt, Ielka van der Sluis, and Kees van Deemter. 2007. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 2007)*, pages 49–56.

Albert Gatt, Anja Belz, and Eric Kow. 2008. The TUNA Challenge 2008: Overview and evaluation results. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 198–206.

Albert Gatt, Anja Belz, and Eric Kow. 2009. The TUNA-REG Challenge 2009: Overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 174–182.

Pablo Gervás, Raquel Hervás, and Carlos Léon. 2008. NIL-UCM: Most-frequent-value-first attribute selection and best-scoring-choice realization. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 215–218.

John Kelleher. 2007. DIT - frequency based incremental attribute selection for GRE. In *Proceedings of the MT Summit XI Workshop Using Corpora for Natural Language Generation: Language Generation and Machine Translation (UCNLG+MT)*, pages 90–92.

Ruud Koolen and Emiel Krahmer. 2010. The D-TUNA corpus: A Dutch dataset for the evaluation of referring expression generation algorithms. In *Proceedings of the 7th international conference on Language Resources and Evaluation (LREC 2010)*.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.

Emiel Krahmer, Mariët Theune, Jette Viethen, and Iris Hendrickx. 2008. GRAPH: The costs of redundancy in referring expressions. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 227–229.

Philipp Spanger, Takehiro Kurosawa, and Takenobu Tokunaga. 2008. On "redundancy" in selecting attributes for generating referring expressions. In *COLING 2008: Companion volume: Posters*, pages 115–118.

Mariët Theune, Ruud Koolen, and Emiel Krahmer. 2010. Cross-linguistic attribute selection for REG: Comparing Dutch and English. In *Proceedings of the 6th International Natural Language Generation Conference (INLG 2010)*, pages 174–182.

Kees van Deemter, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 130–132.

Jette Viethen and Robert Dale. 2010. Speaker-dependent variation in content selection for referring expression generation. In *Proceedings of the 8th Australasian Language Technology Workshop*, pages 81–89.

Jette Viethen, Robert Dale, Emiel Krahmer, Mariët Theune, and Pascal Touset. 2008. Controlling redundancy in referring expressions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 239–246.