

Cross-linguistic attribute selection for REG: Comparing Dutch and English

Mariët Theune

University of Twente
The Netherlands

M.Theune@utwente.nl

Ruud Koolen

Tilburg University
The Netherlands

R.M.F.Koolen@uvt.nl

Emiel Krahmer

Tilburg University
The Netherlands

E.J.Krahmer@uvt.nl

Abstract

In this paper we describe a cross-linguistic experiment in attribute selection for referring expression generation. We used a graph-based attribute selection algorithm that was trained and cross-evaluated on English and Dutch data. The results indicate that attribute selection can be done in a largely language independent way.

1 Introduction

A key task in natural language generation is referring expression generation (REG). Most work on REG is aimed at producing 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). The first step in generating such descriptions is attribute selection: choosing a number of attributes that uniquely characterize the target object. In the next step, realization, the selected attributes are expressed in natural language. Here we focus on the attribute selection step. We investigate to which extent attribute selection can be done in a language independent way; that is, we aim to find out if attribute selection algorithms trained on data from one language can be successfully applied to another language. The languages we investigate are English and Dutch.

Many REG algorithms require training data, before they can successfully be applied to generate references in a particular domain. The Incremental Algorithm (Dale and Reiter, 1995), for example, assumes that certain attributes are more preferred than others, and it is assumed that determining the preference order of attributes is an empirical matter that needs to be settled for each new domain. The graph-based algorithm (Krahmer et al., 2003), to give a second example, similarly assumes that certain attributes are preferred (are

“cheaper”) than others, and that data are required to compute the attribute-cost functions.

Traditional text corpora have been argued to be of restricted value for REG, since these typically are not “semantically transparent” (van Deemter et al., 2006). Rather what seems to be needed is data collected from human participants, who produce referring expressions for specific targets in settings where all properties of the target *and* its distractors are known. Needless to say, collecting and annotating such data takes a lot of time and effort. So what to do if one wants to develop a REG algorithm for a new language? Would this require a new data collection, or could existing data collected for a *different* language be used? Clearly, linguistic realization is language dependent, but to what extent is attribute selection language dependent? This is the question addressed in this paper.

Below we describe the English and Dutch corpora used in our experiments (Section 2), the graph-based algorithm we used for attribute selection (Section 3), and the corpus-based attribute costs and orders used by the algorithm (Section 4). We present the results of our cross-linguistic attribute selection experiments (Section 5) and end with a discussion and conclusions (Section 6).

2 Corpora

2.1 English: the TUNA corpus

For English data, we used the TUNA corpus of object descriptions (Gatt et al., 2007). This corpus was created by presenting the participants in an on-line experiment with a visual scene consisting of seven objects and asking them to describe one of the objects, the target, in such a way that it could be uniquely identified. There were two experimental conditions: in the +LOC condition, the participants were free to describe the target object using any of its properties, including its location on the screen, whereas in the -LOC condition they

were discouraged (but not prevented) from mentioning object locations. The resulting object descriptions were annotated using XML and combined with an XML representation of the visual scene, listing all objects and their properties in terms of attribute-value pairs. The TUNA corpus is split into two domains: one with descriptions of furniture and one with descriptions of people.

The TUNA corpus was used for the comparative evaluation of REG systems in the TUNA Challenges (2007-2009). For our current experiments, we used the TUNA 2008 Challenge training and development sets (Gatt et al., 2008) to train and evaluate the graph-based algorithm on.

2.2 Dutch: the D-TUNA corpus

For Dutch, we used the D(utch)-TUNA corpus of object descriptions (Koolen and Kraemer, 2010). The collection of this corpus was inspired by the TUNA experiment described above, and was done using the same visual scenes. There were three conditions: text, speech and face-to-face. The text condition was a replication (in Dutch) of the TUNA experiment: participants typed identifying descriptions of target referents, distinguishing them from distractor objects in the scene. In the other two conditions participants produced spoken descriptions for an addressee, who was either visible to the speaker (face-to-face condition) or not (speech condition). The resulting descriptions were annotated semantically using the XML annotation scheme of the English TUNA corpus.

The procedure in the D-TUNA experiment differed from that used in the original TUNA experiment in two ways. First, the D-TUNA experiment used a laboratory-based set-up, whereas the TUNA study was conducted on-line in a relatively uncontrolled setting. Second, participants in the D-TUNA experiment were completely prevented from mentioning object locations.

3 Graph-based attribute selection

For attribute selection, we use the graph-based algorithm of Kraemer et al. (2003), one of the highest scoring attribute selection methods in the TUNA 2008 Challenge (Gatt et al. (2008), table 11). In this approach, a visual scene with target and distractor objects is represented as a labelled directed graph, in which the objects are modelled as nodes and their properties as looping edges on the corresponding nodes. To select the

attributes for a distinguishing description, the algorithm searches for a subgraph of the scene graph that uniquely refers to the target referent. Starting from the node representing the target, it performs a depth-first search over the edges connected to the subgraph found so far. The algorithm’s output is the cheapest distinguishing subgraph, given a particular *cost function* that assigns costs to attributes.

By assigning zero costs to some attributes, e.g., the type of an object, the human tendency to mention redundant attributes can be mimicked. However, as shown by Viethen et al. (2008), merely assigning zero costs to an attribute is not a sufficient condition for inclusion; if the graph search terminates before the free attributes are tried, they will not be included. Therefore, the order in which attributes are tried must be explicitly controlled.

Thus, when using the graph-based algorithm for attribute selection, two things must be specified: (1) the cost function, and (2) the order in which the attributes should be searched. Both can be based on corpus data, as described in the next section.

4 Costs and orders

For our experiments, we used the graph-based attribute selection algorithm with two types of cost functions: Stochastic costs and Free-Naïve costs. Both reflect (to a different extent) the relative attribute frequencies found in a training corpus: the more frequently an attribute occurs in the training data, the cheaper it is in the cost functions.

Stochastic costs are directly based on the attribute frequencies in the training corpus. They are derived by rounding $-\log_2(P(v))$ to the first decimal and multiplying by 10, where $P(v)$ is the probability that attribute v occurs in a description, given that the target object actually has this property. The probability $P(v)$ is estimated by determining the frequency of each attribute in the training corpus, relative to the number of target objects that possess this attribute. Free-Naïve costs more coarsely reflect the corpus frequencies: very frequent attributes are “free” (cost 0), somewhat frequent attributes have cost 1 and infrequent attributes have cost 2. Both types of cost functions are used in combination with a stochastic ordering, where attributes are tried in the order of increasing stochastic costs.

In total, four cost functions were derived from the English corpus data and four cost functions derived from the Dutch corpus data. For each lan-

guage, we had two Stochastic cost functions (one for the furniture domain and one for the people domain), and two Free-Naïve cost functions (idem), giving eight different cost functions in total. For each language we determined two attribute orders to be used with the cost functions: one for the furniture domain and one for the people domain.

4.1 English costs and order

For English, we used the Stochastic and Free-Naïve cost functions and the stochastic order from Krahmer et al. (2008). The Stochastic costs and order were derived from the attribute frequencies in the combined training and development sets of the TUNA 2008 Challenge (Gatt et al., 2008), containing 399 items in the furniture domain and 342 items in the people domain. The Free-Naïve costs are simplified versions of the stochastic costs. “Free” attributes are TYPE in both domains, COLOUR for the furniture domain and HASBEARD and HASGLASSES for the people domain. Expensive attributes (cost 2) are X- and Y-DIMENSION in the furniture domain and HASSUIT, HASSHIRT and HASTIE in the people domain. All other attributes have cost 1.

4.2 Dutch costs and order

The Dutch Stochastic costs and order were derived from the attribute frequencies in a set of 160 items (for both furniture and people) randomly selected from the text condition in the D-TUNA corpus. Interestingly, our Stochastic cost computation method led to an assignment of 0 costs to the COLOUR attribute in the furniture domain, thus enabling the Dutch Stochastic cost function to include colour as a redundant property in the generated descriptions. In the English stochastic costs, none of the attributes are free. Another difference is that in the furniture domain, the Dutch stochastic costs for ORIENTATION attributes are much lower than the English costs (except with value FRONT); in the people domain, the same holds for attributes such as HASSUIT and HASTIE. These cost differences, which are largely reflected in the Dutch Free-Naïve costs, do not seem to be caused by differences in expressibility, i.e., the ease with which the attributes can be expressed in the two languages (Koolen et al., 2010); rather, they may be due to the fact that the human descriptions in D-TUNA do not include any DIMENSION attributes.

Language		Furniture		People	
Training	Test	Dice	Acc.	Dice	Acc.
Dutch	Dutch	0.92	0.63	0.78	0.28
	English	0.83	0.55	0.73	0.29
English	Dutch	0.87	0.58	0.75	0.25
	English	0.67	0.29	0.67	0.24

Table 1: Evaluation results for stochastic costs.

Language		Furniture		People	
Training	Test	Dice	Acc.	Dice	Acc.
Dutch	Dutch	0.94	0.70	0.78	0.28
	English	0.83	0.55	0.73	0.29
English	Dutch	0.94	0.70	0.78	0.28
	English	0.83	0.55	0.73	0.29

Table 2: Evaluation results for Free-Naïve costs.

5 Results

All cost functions were applied to both Dutch and English test data. As Dutch test data, we used a set of 40 furniture items and a set of 40 people items, randomly selected from the text condition in the D-TUNA corpus. These items had not been used for training the Dutch cost functions. As English test data, we used a subset of the TUNA 2008 development set (Gatt et al., 2008). To make the English test data comparable to the Dutch ones, we only included items from the -LOC condition (see Section 2.1). This resulted in 38 test items for the furniture domain, and 38 for the people domain.

Tables 1 and 2 show the results of applying the Dutch and English cost functions (with Dutch and English attribute orders respectively) to the Dutch and English test data. The evaluation metrics used, Dice and Accuracy (Acc.), both evaluate human-likeness by comparing the automatically selected attribute sets to those in the human test data. Dice is a set-comparison metric ranging between 0 and 1, where 1 indicates a perfect match between sets. Accuracy is the proportion of system outputs that exactly match the corresponding human data. The results were computed using the ‘teval’ evaluation tool provided to participants in the TUNA 2008 Challenge (Gatt et al., 2008).

To determine significance, we applied repeated measures analyses of variance (ANOVA) to the evaluation results, with three within factors: *training language* (Dutch or English), *cost function* (Stochastic or Free-Naïve), and *domain* (furniture or people), and one between factor representing *test language* (Dutch or English).

An overall effect of cost function shows that the Free-Naïve cost functions generally perform better

than the Stochastic cost functions (Dice: $F(1,76) = 34.853$, $p < .001$; Accuracy: $F(1,76) = 13.052$, $p = .001$). Therefore, in the remainder of this section we mainly focus on the results for the Free-Naïve cost functions (Table 2).

As can be clearly seen in Table 2, Dutch and English Free-Naïve cost functions give almost the same scores in both the furniture and the people domain, when applied to the same test language. The English Free-Naïve cost function performs slightly better than the Dutch one on the Dutch people data, but this difference is not significant.

An overall effect of test language shows that the cost functions (both Stochastic and Free-Naïve) generally give better Dice results on the Dutch data than for the English data (Dice: $F(1,76) = 7.797$, $p = .007$). In line with this, a two-way interaction between test language and training language (Dice: $F(1,76) = 6.870$, $p = .011$) shows that both the Dutch and the English cost functions perform better on the Dutch data than on the English data. However, the overall effect of test language did not reach significance for Accuracy, presumably due to the fact that the Accuracy scores on the English people data are slightly higher than those on the Dutch people data.

Finally, the cost functions generally perform better in the furniture domain than in the people domain (Dice: $F(1,76) = 10.877$, $p = .001$; Accuracy: $F(1,76) = 16.629$, $p < .001$).

6 Discussion

The results of our cross-linguistic attribute selection experiments show that Free-Naïve cost functions, which only roughly reflect the attribute frequencies in the training corpus, have an overall better performance than Stochastic cost functions, which are directly based on the attribute frequencies. This holds across the two languages we investigated, and corresponds with the findings of Krahrmer et al. (2008), who compared Stochastic and Free-Naïve functions that were trained and evaluated on English data only. The difference in performance is probably due to the fact that Free-Naïve costs are less sensitive to the specifics of the training data (and are therefore more generally applicable) and do a better job of mimicking the human tendency towards redundancy.

Moreover, we found that Free-Naïve cost functions trained on different languages (English or Dutch) performed equally well when tested on the

same data (English or Dutch), in both the furniture and people domain. This suggests that attribute selection can in fact be done in a language independent way, using cost functions that have been derived from corpus data in one language to perform attribute selection for another language.

Our results did show an effect of test language on performance: both English and Dutch cost functions performed better when tested on the Dutch D-TUNA data than on the English TUNA data. However, this difference does not seem to be caused by language-specific factors but rather by the quality of the respective test sets. Although the English test data were restricted to the -LOC condition, in which using DIMENSION attributes was discouraged, still more than 25% of the English test data (both furniture and people) included one or more DIMENSION attributes, which were never selected for inclusion by either the English or the Dutch Free-Naïve cost functions. The Dutch test data, on the other hand, did not include any DIMENSION attributes. In addition, the English test data contained more non-unique descriptions of target objects than the Dutch data, in particular in the furniture domain. These differences may be due to the fact that data collection was done in a more controlled setting for D-TUNA than for TUNA. In other words, the seeming effect of test language does not contradict our main conclusion that attribute selection is largely language independent, at least for English and Dutch.

The success of our cross-linguistic experiments may have to do with the fact that English and Dutch hardly differ in the expressibility of object attributes (Koolen et al., 2010). To determine the full extent to which attribute selection can be done in a language-dependent way, additional experiments with less similar languages are necessary.

Acknowledgements

We thank the TUNA Challenge organizers for the English data and the evaluation tool used in our experiments; Martijn Goudbeek for helping with the statistical analysis; and Pascal Touset, Ivo Brugman, Jette Viethen, and Iris Hendrickx for their contributions to the graph-based algorithm. This research is part of the VICI project ‘Bridging the gap between psycholinguistics and computational linguistics: the case of referring expressions’, funded by the Netherlands Organization for Scientific Research (NWO Grant 277-70-007).

References

- R. Dale and E. Reiter. 1995. Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- A. Gatt, I. van der Sluis, and K. 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.
- A. Gatt, A. Belz, and E. 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.
- R. Koolen and E. Krahrmer. 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)*.
- R. Koolen, A. Gatt, M. Goudbeek, and E. Krahrmer. 2010. Overspecification in referring expressions: Causal factors and language differences. Submitted.
- E. Krahrmer, S. van Erk, and A. Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- E. Krahrmer, M. Theune, J. Viethen, and I. 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.
- K. van Deemter, I. I. van der Sluis, and A. 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.
- J. Viethen, R. Dale, E. Krahrmer, M. Theune, and P. Tousek. 2008. Controlling redundancy in referring expressions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 239–246.