Referring Expression Generation Challenge 2008 DIT System Descriptions

J.D. Kelleher
School of Computing
Dublin Institute of Technology
john.kelleher@comp.dit.ie

B. Mac Namee
School of Computing
Dublin Institute of Technology
brian.macnamee@comp.dit.ie

1 Task 1: Attribute Selection

This section describes the two systems developed at DIT for the attribute selection track of the REG 2008 challenge. Both of theses systems use an incremental greedy search to generate descriptions, similar to the incremental algorithm described in (Dale and Reiter, 1995). The output of these incremental algorithms are, to a large extent, determined by the order in which the algorithm tests the target object's attributes for inclusion in the description. Indeed, the major difference between the two systems described in this section is the mechanism used to order the attributes for inclusion.

1.1 DIT-FBI System

The DIT-FBI system selects the next attribute to be tested for inclusion in the description by ordering each attribute based on its frequency in the subset of the training corpus that is defined by the test trial's domain (i.e., furniture versus people) and condition (+LOC versus -LOC). Attributes are selected in descending order of frequency (i.e. the attribute that occurred most frequently in the relevant subset of the training corpus is selected first). Where two or more attributes have the same frequency of occurrence the first attribute found with that frequency is selected. The *type* attribute is always included in the description. Other attributes are included in the description if they exclude at least 1 distractor from the set of distractors that fulfil the description generated prior to that attribute's selection.

The mapping from qualitative linguistics descriptions, such as middle or centre, to the TUNA corpus' quantitative location attribute values, i.e, *x-dimension* and *y-dimension*, can result in both the *x-dimension* and *y-dimension* attributes being included in the target set. These cases, where both the dimensional attributes are required, are difficult to capture because each of the dimensional attributes would be sufficiently discriminative to result in a distinguishing description simply by their lone inclusion. As a result, a rule was put in place whereby if we have included either of the dimensional attributes we include the other dimensional attribute if the included one refers to the center of the display (i.e., x=3, y=2).

The algorithm terminates when a distinguishing description has been generated (i.e., all the distractors have been excluded) or when all of the target's attributes have been tested for inclusion in the description. Table 1 lists the results for the system.

	Furniture	People	Both
Dice	0.816	0.702	0.763
MASI	0.606	.452	0.535
Accuracy	37	17	54
Minimality	80	68	148
Uniqueness	0	0	0

Table 1: Results for furniture, people and both domains.

1.2 DIT-TVAS System

In the DIT-TVAS system the selection of the next target attribute to test for inclusion in the description is based on the prior probability of a target's attribute *with a particular value* being used to describe the target, given that the target has a particular attribute with that particular value. This prior is computed by counting the number of trials in the training corpus where the *target description* included a particular attribute-value pair and dividing this count by the number of trials in the training corpus where the target's properties listed a particular attribute-value pair.

For example, there are *143* trials in the Reg-08-Challenge training corpus where the target's type attribute had the value *chair*, and in *134* of these trials the description of the target included the attribute value pair *type-chair*. As a result, the atribute-value pair *type-chair* has a prior probability of being used to describe the target, given that the target properties contain this attribute-value combination, of $\frac{134}{143} \approx 0.937$. Table 2 lists the priors for each attribute-value combination in the furniture corpus and Table 3 lists the priors for each attribute-value combination in the people corpus (for space reasons these tables do not include the priors for the *x-dimension*, *y-dimension* and *other* attributes-value pairs). Given these tables and a test trial, the next attribute-value pair to be tested for inclusion in the description of the test target is the attibute-value with the highest prior that has not already been tested for inclusion and that the target object fulfils.

As is evident from Table 2 and Table 3, there is no significant difference between the priors of some attribute-value pairs. For example, in the furniture domain *orientation=left* and *hasShirt=true* have priors of 0.023 and 0.022 respectively. In order to avoid situations where a non-significant difference in priors unduly biases the system toward the inclusion of a particular attribute-value pair, each time an attribute-vaule pair has been selected for testing the DIT-TVAS system checks whether there are any other attribute-value pairs that have not been previously tested for inclusion in the description of the target and whose prior is within 5% of the prior of the attribute-value that has been selected for testing. In cases where this test returns one or more attribute-value pairs, the system uses the attribute-value pair whose inclusion would exclude the most amount of distractors. Finally, if there is a tie between one or more attribute-value pairs with respect to distractor exclusion this is resolved by slecting the attribute-value pair with the highest prior. Table 4 lists the results for the system.

Attribute	VALUE	Sel	Occur	Prior
TYPE	fan	41	42	0.976
TYPE	chair	134	143	0.937
TYPE	sofa	43	48	0.896
COLOUR	green	35	40	0.875
COLOUR	blue	75	86	0.872
COLOUR	red	82	96	0.854
TYPE	desk	73	86	0.849
COLOUR	grey	81	97	0.835
ORIENTATION	back	25	51	0.490
SIZE	small	56	130	0.431
ORIENTATION	front	31	86	0.360
SIZE	large	61	189	0.324
ORIENTATION	left	22	86	0.256
ORIENTATION	right	28	96	0.292

Table 2: Prior's for each attribute-value pair in the furniture domain. Sel: how often an attribute-value pair was included in a description; Occur: how often an attribute-vaule pair appeared in targets in the training corpus. Prior's listed to three decimal places.

Attribute	VALUE	Sel	Occur	Prior
TYPE	person	225	274	0.821
hasBeard	true	123	181	0.680
hasGlasses	true	117	184	0.636
hasSuit	true	4	94	0.43
hasHair	true	36	233	0.155
hasHair	false	6	41	0.146
AGE	old	15	132	0.114
ORIENTATION	right	2	44	0.045
ORIENTATION	front	4	143	0.028
ORIENTATION	left	2	87	0.023
hasShirt	true	3	136	0.022
hasTie	true	2	94	0.021
AGE	young	2	142	0.014
hasShirt	false	0	138	0
hasBeard	false	0	93	0
hasGlasses	false	0	90	0
hasTie	false	0	180	0
hasSuit	false	0	180	0

Table 3: Prior's for each attribute-value pair in the furniture domain. Sel: how often an attribute-value pair was included in a description; Occur: how often an attribute-vaule pair appeared in targets in the training corpus. Prior's listed to three decimal places.

	Furniture	People	Both
Dice	0.778	0.709	0.746
MASI	0.540	.426	0.488
Accuracy	33	15	48
Uniqueness	80	68	148
Minimality	0	0	0

Table 4: Results for furniture, people and both domains.

2 Task 2: Realisation

This section describes the two systems developed at DIT for the realisation track of the REG 2008 challenge. The DIT-CBSR system, Section 2.1, uses a case-based reasoning approach to realization, which (Daelemans and van den Bosch, 2005) have recently argued is an appropriate machine learning approach to natural language processing. The DIT-RBR system, Section 2.2, uses a set of hand-crafted domain-specific rules to generate descriptions.

2.1 DIT-CBSR System

Cased-Based Reasoning attempts to use a history of past problems and their solutions to solve newly arising problems. The solution to a new problem is generated by finding the problem in the set of training problems the system has previously seen (i.e. the *case base*) which most closely matches it and adapting its solution.

The DIT-CBSR system uses a relatively simple case matching algorithm. When a new trial requires sentence generation it's attribute set is matched against all of the cases in the training set to determine which cases is matches most closely. This matching firstly considers only attributes and their values. There are three kinds of matches that can arise from this process:

- **Perfect match:** the attributes used in both the query case and the case from the case-base match perfectly as do their values.
- **Partial match:** the attribute used by both the query case and the case from the case-base match perfectly, but the attribute values do not match.
- **No match:** no member of the case-base has a list of attributes used that match those required by the query case.

Slightly different actions are taken depending on the type of match achieved. These are as follows.

Perfect Match Perfect matches are the easiest to deal with as little effort is required in order to produce a useful sentence. In fact, if only one perfect match is found then that trial's word string is used, unedited, as the generated sentence. However, things become a little more interesting if more than one case in the case-base matches the query case perfectly (remembering that the match is only based on the attributes used and their values). In this case, the list of matches is first trimmed of any cases that are not based on the same image as the query case, as long as this does not remove all cases. If this does remove all cases we revert to the original set of matching cases. In either instance, the word strings in the set of remaining matching cases are considered to determine if there are any duplicates. If there are, the word string that appears most frequently is used as the generated sentence. If there are no duplicates, the shortest word string in the set is chosen.

Partial Match Partial matches occur when there is no example in the case-base for which all of the attribute values are the same as those of the query case. However, there are some case whose attributes match the query, but whose attribute values are different. This set of cases is sub-divided based on the number of attribute values in each case that match those in the query case. This results in a set of cases that share the highest match score. From this set all of those cases that are not based on the same image as the query case are removed, as long as this does not completely empty the set. If this trimming would completely empty the set it is not performed. The trial with the shortest word string from the set of remaining candidate matches is selected. The word string associated with the selected trial needs to be modified to account for the disparity between its attribute values and those of the query case. This modification is done by replacing all substrings in the selected case's word string that arise from attribute values not matching those in the query case with the substring that is most commonly associated in the training corpus with the query case's attribute value. All of these substrings are identified using the annotated word string element present in each trial.

No Match When no match is found a simple rule-based realiser is used to construct a sentence matching the attribute value set of the query case. The rules used by the realiser are based on the most common strings found in the corpus for each attribute value pair.

Table 5 lists the results for the system.

	Furniture	People	Both
String-edit distance	3.95	4.81	4.34
Accuracy score	12	6	18

Table 5: Results for furniture, people and both domains.

2.2 DIT-RBR System

In Section 2.1 we noted that if no match was found in the case-base a simple rule-based realiser was used. This rule-based realiser, the DIT-RBR system, uses a sequence of `IF-THEN` rules based on a study of the frequencies and order of the phrases used to realise specific attribute-value pairs in the training corpus. Theses phrase are easily extracted from the annotated word-string xml element. The great advantage of this algorithm is that it always able to return a string given an input. However, the rule-set is specific to the task and would not generalise as well as the DIT-CBSR system. Due to space restrictions we do not list the rules used by the system. Table 6 lists the results for the system.

	Furniture	People	Both
String edit distance	3.613	4.132	3.851
Accuracy	11	3	14

Table 6: Results for furniture, people and both domains.

3 Task 3: Referring Expression Generation

This section describes describes our approach to task 3 of REG Challenge 2008. Each of the systems described in this section simply chains together DIT solutions to task 1 and task 2.

3.1 DIT-FBI-CBSR System

The DIT-FBI-CBSR system chains together the DIT-FBI attribute selection system, described in 1.1, and the DIT-CBSR system, described in Section 2.1. Table 7 lists the results for the system.

	Furniture	People	Both
String-edit distance	4.45	5.162	4.777
Accuracy score	7	1	8

Table 7: Results for furniture, people and both domains.

3.2 DIT-TVAS-RBR Task 3 System

The DIT-TVAS-RBR system chains together the DIT-TVAS attribute selection system, described in Section 1.2, and the DIT-RBR realiser, described in Section 2.2. Table 8 lists the results for the system.

	Furniture	People	Both
String-edit distance	4.725	5.178	4.905
Accuracy score	4	0	4

Table 8: Results for furniture, people and both domains.

References

Walter Daelemans and Antal van den Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press.

R. Dale and E. Reiter. 1995. Computatinal interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 18:233–263.