Structuring Knowledge for Reference Generation: A Clustering Algorithm

Albert Gatt
Department of Computing Science
University of Aberdeen
Scotland, United Kingdom
agatt@csd.abdn.ac.uk

Abstract

This paper discusses two problems that arise in the Generation of Referring Expressions: (a) numeric-valued attributes, such as size or location; (b) perspective-taking in reference. Both problems, it is argued, can be resolved if some structure is imposed on the available knowledge prior to content determination. We describe a clustering algorithm which is sufficiently general to be applied to these diverse problems, discuss its application, and evaluate its performance.

1 Introduction

The problem of Generating Referring Expressions (GRE) can be summed up as a search for the properties in a knowledge base (KB) whose combination uniquely distinguishes a set of referents from their distractors. The content determination strategy adopted in such algorithms is usually based on the assumption (made explicit in Reiter (1990)) that the space of possible descriptions is partially ordered with respect to some principle(s) which determine their adequacy. Traditionally, these principles have been defined via an interpretation of the Gricean maxims (Dale, 1989; Reiter, 1990; Dale and Reiter, 1995; van Deemter, 2002)[1]. However, little attention has been paid to contextual or intentional influences on attribute selection (but cf. Jordan and Walker (2000); Krahmer and Theune (2002)). Furthermore, it is often assumed that all relevant knowledge about domain objects is represented in the database in a format (e.g. attribute-value pairs) that requires no further processing.

This paper is concerned with two scenarios which raise problems for such an approach to GRE:

1. *Real-valued attributes*, e.g. size or spatial coordinates, which represent continuous dimensions. The utility of such attributes depends on whether a set of referents have values that are 'sufficiently close' on the given dimension, and 'sufficiently distant' from those of their distractors. We discuss this problem in §2.
2. *Perspective-taking* The contextual appropriateness of a description depends on the perspective being taken in context. For instance, if it is known of a referent that it is a *teacher*, and a *sportsman*, it is better to talk of *the teacher* in a context where another referent has been introduced as *the student*. This is discussed further in §3.

Our aim is to motivate an approach to GRE where these problems are solved by pre-processing the information in the knowledge base, prior to content determination. To this end, §4 describes a clustering algorithm and shows how it can be applied to these different problems to structure the KB prior to GRE.

2 Numeric values: The case of location

Several types of information about domain entities, such as gradable properties (van Deemter, 2000) and physical location, are best captured by real-valued attributes. Here, we focus on the example of location as an attribute taking a tuple of values which jointly determine the position of an entity.

The ability to distinguish groups is a well-established feature of the human perceptual apparatus (Wertheimer, 1938; Treisman, 1982). Representing salient groups can facilitate the task of excluding distractors in the search for a referent. For instance, the set of referents marked as the intended referential target in Figure 1 is easily distinguishable as a group and warrants the use of a spatial description such as *the objects in the top left corner*, possibly with a collective predicate, such as *clustered* or *gathered*. In case of reference to a subset of the marked set, although location would be insufficient to distinguish the targets, it would reduce the distractor set and facilitate reference resolution[2].

In GRE, an approach to spatial reference based on grouping has been proposed by Funakoshi *et al.*

[1] For example, the Gricean Brevity maxim (Grice, 1975) has been interpreted as a directive to find the shortest possible description for a given referent

[2] Location has been found to significantly facilitate resolution, even when it is logically redundant (Arts, 2004)

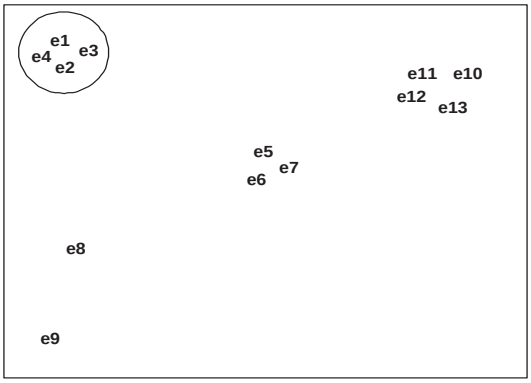

Figure 1: Spatial Example

(2004). Given a domain and a target referent, a *sequence of groups* is constructed, starting from the largest group containing the referent, and recursively narrowing down the group until only the referent is identified. The entire sequence is then rendered linguistically. The algorithm used for identifying perceptual groups is the one proposed by Thorisson (1994), the core of which is a procedure which takes as input a list of pairs of objects, ordered by the distance between the entities in the pairs. The procedure loops through the list, finding the greatest difference in distance between two adjacent pairs. This is determined as a cutoff point for group formation. Two problems are raised by this approach:

P1 **Ambiguous clusters** A domain entity can be placed in more than one group. If, say, the input list is $\langle \{a, b\}, \{c, e\}, \{a, f\} \rangle$ and the greatest difference after the first iteration is between $\{c, e\}$ and $\{a, f\}$, then the first group to be formed will be $\{a, b, c, e\}$ with $\{a, f\}$ likely to be placed in a different group after further iterations. This may be confusing from a referential point of view. The problem arises because grouping or clustering takes place on the basis of *pairwise* proximity or distance. This problem can be partially circumvented by identifying groups on several perceptual dimensions (e.g. spatial distance, colour, and shape) and then seeking to merge identical groups determined on the basis of these different qualities (see Thorisson (1994)). However, the grouping strategy can still return groups which do not conform to human perceptual principles. A better strategy is to base clustering on the Nearest Neighbour Principle, familiar from computational geometry (Prepaarata and Shamos, 1985), whereby elements are clustered with their nearest neighbours, given a distance function. The solution offered below is based on this principle.

P2 **Perceptual proximity** Absolute distance is not sufficient for cluster identification. In Figure 1, for example, the pairs $\{e_1, e_2\}$ and $\{e_5, e_6\}$ could easily be consecutively ranked, since the distance between e_1 and e_2 is roughly equal to that between e_5 and e_6. However, they would not naturally be clustered together by a human observer, because grouping of objects also needs to take into account the position of the surrounding elements. Thus, while e_1 is as far away from e_2 as e_5 is from e_6, there are elements which are closer to $\{e_1, e_2\}$ than to $\{e_5, e_6\}$.

The proposal in §4 represents a way of getting around these problems, which are expected to arise in any kind of domain where the information given is the pairwise distance between elements. Before turning to the framework, we consider another situation in GRE where the need for clustering could arise.

3 Perspectives and semantic similarity

In real-world discourse, entities can often be talked about from different points of view, with speakers bringing to bear world and domain-specific knowledge to select information that is relevant to the current topic. In order to generate coherent discourse, a generator should ideally keep track of how entities have been referred to, and maintain consistency as far as possible.

	type	**profession**	**nationality**
e_1	man	student	englishman
e_2	woman	teacher	italian
e_3	man	chef	greek

Table 1: Semantic Example

Suppose e_1 in Table 1 has been introduced into the discourse via the description *the student* and the next utterance requires a reference to e_2. Any one of the three available attributes would suffice to distinguish the latter. However, a description such as *the woman* or *the italian* would describe this entity from a different point of view relative to e_1. By hypothesis, *the teacher* is more appropriate, because the property ascribed to e_2 is more similar to that ascribed to e_1.

A similar case arises with plural disjunctive descriptions of the form $\lambda x[p(x) \vee q(x)]$, which are usually realised as coordinate constructions of the form *the N'*$_1$ *and the N'*$_2$. For instance a reference to $\{e_1, e_2\}$ such as *the woman and the student*, or *the englishman and the teacher*, would be odd, compared to the alternative *the student and the teacher*. The latter describes these entities under the same perspective. Note that 'consistency' or 'similarity' is not guaranteed simply by attempting to use values of the same attribute(s) for a given set of referents. The description *the student*

and the chef for $\{e_1, e_3\}$ is relatively odd compared to the alternative *the englishman and the greek*. In both kinds of scenarios, a GRE algorithm that relied on a rigid preference order could not guarantee that a coherent description would be generated every time it was available.

The issues raised here have never been systematically addressed in the GRE literature, although support for the underlying intuitions can be found in various quarters. Kronfeld (1989) distinguishes between *functionally* and *conversationally* relevant descriptions. A description is functionally relevant if it succeeds in distinguishing the intended referent(s), but conversational relevance arises in part from implicatures carried by the use of attributes in context. For example, describing e_1 as *the student* carries the (Gricean) implicature that the entity's academic role or profession is somehow relevant to the current discourse. When two entities are described using contrasting properties, say *the student and the italian*, the listener may find it harder to work out the relevance of the contrast. In a related vein, Aloni (2002) formalises the appropriateness of an answer to a question of the form *Wh x?* with reference to the 'conceptual covers' or perspectives under which *x* can be conceptualised, not all of which are equally relevant given the hearer's information state and the discourse context.

With respect to plurals, Eschenbach *et al.* (1989) argue that the generation of a plural anaphor with a split antecedent is more felicitous when the antecedents have something in common, such as their ontological category. This constraint has been shown to hold psycholinguistically (Kaup et al., 2002; Koh and Clifton, 2002; Moxey et al., 2004). Gatt and van Deemter (2005a) have shown that people's perception of the adequacy of plural descriptions of the form, *the* N_1 *and (the)* N_2 is significantly correlated with the semantic similarity of N_1 and N_2, while singular descriptions are more likely to be aggregated into a plural if semantically similar attributes are available (Gatt and Van Deemter, 2005b).

The two kinds of problems discussed here could be resolved by pre-processing the KB in order to identify available perspectives. One way of doing this is to group available properties into clusters of semantically similar ones. This requires a well-defined notion of 'similarity' which determines the 'distance' between properties in semantic space. As with spatial clustering, the problem is then of how to get from pairwise distance to well-formed clusters or groups, while respecting the principles underlying human perceptual/conceptual organisation. The next section describes an algorithm that aims to achieve this.

4 A framework for clustering

In what follows, we assume the existence of a set of clusters $\mathcal{C}$ in a domain S of objects (entities or properties), to be 'discovered' by the algorithm. We further assume the existence of a *dimension*, which is characterised by a function δ that returns the pairwise distance $\delta(a, b)$, where $\langle a, b\rangle \in S \times S$. In case an attribute is characterised by more than one dimension, say $\langle x, y\rangle$ coordinates in a 2D plane, as in Figure 1, then δ is defined as the Euclidean distance between pairs:

$$\delta = \sqrt{\sum_{\langle x,y\rangle \in D} |x_{ab} - y_{ab}|^2} \tag{1}$$

where D is a tuple of dimensions, $x_{ab} = \delta(a, b)$ on dimension x. δ satisfies the axioms of *minimality* (2a), *symmetry* (2b), and the *triangle inequality* (2c), by which it determines a metric space on S:

$$\delta(a, b) \geq 0 \wedge \big(\delta(a, b) = 0 \leftrightarrow a = b\big) \tag{2a}$$

$$\delta(a, b) = \delta(b, a) \tag{2b}$$

$$\delta(a, b) + \delta(b, c) \geq \delta(a, c) \tag{2c}$$

We now turn to the problems raised in §2. P1 would be avoided by a clustering algorithm that satisfies (3).

$$\bigcap_{C_i \in \mathcal{C}} = \emptyset \tag{3}$$

It was also suggested above that a potential solution to P1 is to cluster using the Nearest Neighbour Principle. Before considering a solution to P2, i.e. the problem of discovering clusters that approximate human intuitions, it is useful to recapitulate the classic principles of perceptual grouping proposed by Wertheimer (1938), of which the following two are the most relevant:

1. *Proximity* The smaller the distance between objects in the cluster, the more easily perceived it is.

2. *Similarity* Similar entities will tend to be more easily perceived as a coherent group.

Arguably, once a numeric definition of (semantic) similarity is available, the Similarity Principle boils down to the Proximity principle, where proximity is defined via a semantic distance function. This view is adopted here. How well our interpretation of these principles can be ported to the semantic clustering problem of §3 will be seen in the following subsections.

To resolve P2, we will propose an algorithm that uses a context-sensitive definition of 'nearest neighbour'. Recall that P2 arises because, while δ is a measure of 'objective' distance on some scale, *perceived*

proximity (resp. distance) of a pair $\langle a, b\rangle$ is contingent not only on $\delta(a, b)$, but also on the distance of a and b from all other elements in S. A first step towards meeting this requirement is to consider, for a given pair of objects, not only the absolute distance (proximity) between them, but also the extent to which they are equidistant from other objects in S. Formally, a measure of perceived proximity $prox(a, b)$ can be approximated by the following function. Let the two sets P_{ab}, D_{ab} be defined as follows:

$$P_{ab} = \{x | x \in S \wedge \delta(x, a) \sim \delta(x, b)\}$$

$$D_{ab} = \{y | y \in S \wedge \delta(y, a) \not\sim \delta(y, b)\}$$

Then:

$$prox(a, b) = F(\delta(a, b), |P_{ab}|, |D_{ab}|) \qquad (4)$$

that is, $prox(a, b)$ is a function of the absolute distance $\delta(a, b)$, the number of elements in $S - \{a, b\}$ which are roughly equidistant from a and b, and the number of elements which are not equidistant. One way of conceptualising this is to consider, for a given object a, the list of all other elements of S, ranked by their distance (proximity) to a. Suppose there exists an object b whose ranked list is similar to that of a, while another object c's list is very different. Then, all other things being equal (in particular, the pairwise absolute distance), a clusters closer to b than does c.

This takes us from a metric, distance-based conception, to a broader notion of the 'similarity' between two objects in a metric space. Our definition is inspired by Tversky's feature-based Contrast Model (1977), in which the similarity of a, b with feature sets A, B is a linear function of the features they have in common and the features that pertain only to A or B, i.e.: $sim(a, b) = f(A \cap B) - f(\overline{A \cap B})$. In (4), the distance of a and b from every other object is the relevant feature.

4.1 Computing perceived proximity

The computation of pairwise perceived proximity $prox(a, b)$, shown in Algorithm 1, is the first step towards finding clusters in the domain.

Following Thorisson (1994), the procedure uses the absolute distance δ to calculate 'absolute proximity' (1.7), a value in $(0, 1)$, with 1 corresponding to $\delta(a, b) = 0$, i.e. identity (cf. axiom (2a)). The procedure then visits each element of the domain, and compares its rank with respect to a and b (1.9–1.13)[3], incrementing a proximity score s (1.10) if the ranks are approximately equal, or a distance score d otherwise (1.12). Approximate equality is determined via a constant k (1.1), which, based on our experiments is set to a tenth the size of S. The procedure returns the ratio of proximity and distance scores, weighted by the absolute proximity $p(a, b)$ (1.15). Algorithm 1 is called for all pairs in $S \times S$ yielding, for each element $a \in S$, a list of elements ordered by their perceived proximity to a. The entity with the highest proximity to a is called its *anchor*. Note that any domain object has one, and only one anchor.

[3]We simplify the presentation by assuming the function $rank(x, a)$ that returns the rank of x with respect to a. In practice, this is achieved by creating, for each element of the input pair, a totally ordered list $\mathcal{L}_a$ such that $\mathcal{L}_a[r]$ holds the set of elements ranked at r with respect to $\delta(x, a)$

Algorithm 1 prox(a,b)

Require: $\delta(a, b)$
Require: k (a constant)
1: $maxD \leftarrow \max_{\langle x,y\rangle \in S \times S} \delta(x, y)$
2: **if** $a = b$ **then**
3: **return** 1
4: **end if**
5: $s \leftarrow 0$
6: $d \leftarrow 0$
7: $p(a, b) \leftarrow 1 - \frac{\delta(a,b)}{maxD}$
8: **for all** $x \in S - \{a, b\}$ **do**
9: **if** $|rank(x, a) - rank(x, b)| \leq k$ **then**
10: $s \leftarrow s + 1$
11: **else**
12: $d \leftarrow d + 1$
13: **end if**
14: **end for**
15: **return** $p(a, b) \times \frac{s}{d}$

4.2 Creating clusters

The procedure $makeClusters(S, Anchors)$, shown in its basic form in Algorithm 2, uses the notion of an anchor introduced above. The rationale behind the algorithm is captured by the following declarative principle, where $C \in \mathcal{C}$ is any cluster, and $anchor(a, b)$ means 'b is the anchor of a':

$$a \in C \wedge anchor(a, b) \rightarrow b \in C \qquad (5)$$

A cluster is defined as the transitive closure of the anchor relation, that is, if it holds that $anchor(a, b)$ and $anchor(b, c)$, then $\{a, b, c\}$ will be clustered together. Apart from satisfying (5), the procedure also induces a partition on S, satisfying (3). Given these primary aims, no attempt is made, once clusters are generated, to further sub-divide them, although we briefly return to this issue in §5. The algorithm initialises a set $Clusters$ to empty (2.1), and iterates through the list of objects S (2.5). For each object a and its anchor b (2.6), it first checks whether they have already been clustered (e.g. if either of them was the anchor of an object visited earlier) (2.7, 2.12). If this is not the case, then a provisional cluster is initialised for each element

Algorithm 2 makeClusters(S, Anchors)

Ensure: $S \neq \emptyset$
1: Clusters $\leftarrow \emptyset$
2: **if** $|S| = 1$ **then**
3: **return** S
4: **end if**
5: **for all** $a \in S$ **do**
6: $b \leftarrow$ Anchors[a]
7: **if** $\exists C \in Clusters : a \in C$ **then**
8: $C_a \leftarrow C$
9: **else**
10: $C_a \leftarrow \{a\}$
11: **end if**
12: **if** $\exists C \in Clusters : b \in C$ **then**
13: $C_b \leftarrow C$
14: $Clusters \leftarrow Clusters - \{C_b\}$
15: **else**
16: $C_b \leftarrow \{b\}$
17: **end if**
18: $C_a \leftarrow C_a \cup C_b$
19: $Clusters \leftarrow Clusters \cup \{C_a\}$
20: **end for**
21: **return** Clusters

(2.10, 2.16). The procedure simply merges the cluster containing a with that of its b (2.18), having removed the latter from the cluster set (2.14).

This algorithm is guaranteed to induce a partition, since no element will end up in more than one group. It does not depend on an ordering of pairs à la Thorisson. However, problems arise when elements and anchors are clustered näively. For instance, if an element is very distant from every other element in the domain, $prox(a, b)$ will still find an anchor for it, and $makeClusters(S, Anchors)$ will place it in the same cluster as its anchor, although it is an outlier. Before describing how this problem is rectified, we introduce the notion of a *family* (F) of elements. Informally, this is a set of elements of S that have the same anchor, that is:

$$\forall a, b \in F : anchor(a, x) \wedge anchor(b, y) \leftrightarrow x = y \quad (6)$$

The solution to the outlier problem is to calculate a *centroid value* for each family found after $prox(a, b)$. This is the average proximity between the common anchor and all members of its family, minus one standard deviation. Prior to merging, at line (2.18), the algorithm now checks whether the proximity value between an element and its anchor falls below the centroid value. If it does, the the cluster containing an object and that containing its anchor are not merged.

4.3 Two applications

The algorithm was applied to the two scenarios described in §2 and §3. In the spatial domain, the algorithm returns groups or clusters of entities, based on their spatial proximity. This was tested on domains like Figure 1 in which the input is a set of entities whose position is defined as a pair of x/y coordinates. Figure 1 illustrates a potential problem with the procedure. In that figure, it holds that $anchor(e_8, e_9)$ and $anchor(e_9, e_8)$, making e_8 and e_9 a *reciprocal pair*. In such cases, the algorithm inevitably groups the two elements, whatever their proximity/distance. This may be problematic when elements of a reciprocal pair are very distant from eachother, in which case they are unlikely to be perceived as a group. We return to this problem briefly in §5.

The second domain of application is the clustering of properties into 'perspectives'. Here, we use the information-theoretic definition of similarity developed by Lin (1998) and applied to corpus data by Kilgarriff and Tugwell (Kilgarriff and Tugwell, 2001). This measure defines the similarity of two words as a function of the likelihood of their occurring in the same grammatical environments in a corpus. This measure was shown experimentally to correlate highly with human acceptability judgments of disjunctive plural descriptions (Gatt and van Deemter, 2005a), when compared with a number of measures that calculate the similarity of word senses in WordNet. Using this as the measure of semantic distance between words, the algorithm returns clusters such as those in Figure 2.

input:	{ waiter, essay, footballer, article, servant, cricketer, novel, cook, book, maid, player, striker, goalkeeper }
output:	
1	{ essay, article, novel, book }
2	{ footballer, cricketer }
3	{ waiter, cook, servant, maid }
4	{ player, goalkeeper, striker }

Figure 2: Output on a Semantic Domain

If the words in Figure 2 represented properties of different entities in the domain of discourse, then the clusters would represent perspectives or 'covers', whose extension is a set of entities that can be talked about from the same point of view. For example, if some entity were specified as having the property *footballer*, and the property *striker*, while another entity had the property *cricketer*, then according to the output of the algorithm, the description *the footballer and the cricketer* is the most conceptually coherent one available. It could be argued that the units of representation

	spatial	semantic
1	0.94	0.58
2	0.86	0.36
3	0.62	0.76
4	0.93	0.52
mean	0.84	0.64

Table 2: Proportion of agreement among participants

in GRE are not words but 'properties' (e.g. values of attributes) which can be realised in a number of different ways (if, for instance, there are a number of synonyms corresponding roughly to the same intension). This could be remedied by defining similarity as 'distance in an ontology'; conversely, properties could be viewed as a set of potential (word) realisations.

5 Evaluation

The evaluation of the algorithm was based on a comparison of its output against the output of human beings in a similar task.

Thirteen native or fluent speakers of English volunteered to participate in the study. The materials consisted of 8 domains, 4 of which were graphical representations of a 2D spatial layout containing 13 points. The pictures were generated by plotting numerical x/y coordinates (the same values are used as input to the algorithm). The other four domains consisted of a set of 13 arbitrarily chosen nouns. Participants were presented with an eight-page booklet with spatial and semantic domains on alternate pages. They were instructed to draw circles around the best clusters in the pictures, or write down the words in groups that were related according to their intuitions. Clusters could be of arbitrary size, but each element had to be placed in exactly one cluster.

5.1 Participant agreement

Participant agreement on each domain was measured using kappa. Since the task did not involve predefined clusters, the set of *unique* groups (denoted G) generated by participants in every domain was identified, representing the set of 'categories' available post hoc. For each domain element, the number of times it occurred in each group served as the basis to calculate the proportion of agreement among participants for the element. The total agreement $P(A)$ and the agreement expected by chance, $P(E)$ were then used in the standard formula

$$k = \frac{P(A) - P(E)}{1 - P(E)}$$

Table 2 shows a remarkable difference between the two domain types, with very high agreement on spatial domains and lower values on the semantic task. The difference was significant ($t = 2.54$, $p < 0.05$). Disagreement on spatial domains was mostly due to the problem of reciprocal pairs, where participants disagreed on whether entities such as e_8 and e_9 in Figure 1 gave rise to a well-formed cluster or not. However, all the participants were consistent with the version of the Nearest Neighbour Principle given in (5). If an element was grouped, it was always grouped with its anchor.

The disagreement in the semantic domains seemed to turn on two cases[4]:

1. *Sub-clusters* Whereas some proposals included clusters such as { *man, woman, boy, girl, infant, toddler, baby, child* } , others chose to group { *infant, toddler, baby,child* } separately.

2. *Polysemy* For example, *liver* was in some cases clustered with { *steak, pizza* } , while others grouped it with items like { *heart, lung* } .

Insofar as an algorithm should capture the whole range of phenomena observed, (1) above could be accounted for by making repeated calls to the Algorithm to subdivide clusters. One problem is that, in case only one cluster is found in the original domain, the same cluster will be returned after further attempts at sub-clustering. A possible solution to this is to redefine the parameter k in Algorithm (1), making the condition for proximity more strict. As for the second observation, the desideratum expressed in (3) may be too strong in the semantic domain, since words can be polysemous. As suggested above, one way to resolve this would be to measure distance between word senses, as opposed to words.

5.2 Algorithm performance

The performance of the algorithm (hereafter the *target*) against the human output was compared to two baseline algorithms. In the spatial domains, we used an implementation of the Thorisson algorithm (Thorisson, 1994) described in §2. In our implementation, the procedure was called iteratively until all domain objects had been clustered in at least one group.

For the semantic domains, the baseline was a simple procedure which calculated the powerset of each domain S. For each subset in $pow(S) - \{\emptyset, S\}$, the procedure calculates the mean pairwise similarity between words, returning an ordered list of subsets. This partial order is then traversed, choosing subsets until all elements had been grouped. This seemed to be a reasonable baseline, because it corresponds to the intuition that the 'best cluster' from a semantic point of view is the one with the highest pairwise similarity among its elements.

[4]The conservative strategy used here probably amplifies disagreements; disregarding clusters which are subsumed by other clusters would control at least for case (1)

The output of the target and baseline algorithms was compared to human output in the following ways:

1. *By item* In each of the eight test domains, an agreement score was calculated for each domain element e (i.e. 13 scores in each domain). Let U_s be the set of distinct groups containing e proposed by the experimental participants, and let U_a be the set of unique groups containing e proposed by the algorithm ($|U_a| = 1$ in case of the target algorithm, but not necessarily for the baselines, since they do not impose a partition). For each pair $\langle U_{a_i}, U_{s_j} \rangle$ of algorithm-human clusters, the agreement score was defined as

$$\frac{|U_{a_i} \cap U_{s_j}|}{|U_{a_i} \cap U_{s_j}| + |\overline{U_{a_i} \cap U_{s_i}}|},$$

   i.e. the ratio of the number of elements on which the human/algorithm agree, and the number of elements on which they do not agree. This returns a number in $(0, 1)$ with 1 indicating perfect agreement. The maximal such score for each entity was selected. This controlled for the possible advantage that the target algorithm might have, given that it, like the human participants, partitions the domain.

2. *By participant* An overall mean agreement score was computed for each participant using the above formula for the target and baseline algorithms in each domain.

Results by item Table 3 shows the mean and modal agreement scores obtained for both target and baseline in each domain type. At a glance, the target algorithm performed better than the baseline on the spatial domains, with a modal score of 1, indicating perfect agreement on 60% of the objects. The situation is different in the semantic domains, where target and baseline performed roughly equally well; in fact, the modal score of 1 accounts for 75% baseline scores.

		target	**baseline**
spatial	mean	0.84	0.72
	mode	1 (60%)	0.67 (40%)
semantic	mean	0.86	0.86
	mode	1 (65%)	1 (75%)

Table 3: Mean and modal agreement scores

Unsurprisingly, the difference between target and baseline algorithms was reliable on the spatial domains ($t = 2.865$, $p < .01$), but not on the semantic domains ($t < 1$, ns). This was confirmed by a one-way Analysis of Variance (ANOVA), testing the effect of algorithm (target/baseline) and domain type (spatial/semantic) on agreement results. There was a significant main effect of domain type ($F = 6.399$, $p = .01$), while the main effect of algorithm was marginally significant ($F = 3.542$, $p = .06$). However, there was a reliable type $\times$ algorithm interaction ($F = 3.624$, $p = .05$), confirming the finding that the agreement between target and human output differed between domain types. Given the relative lack of agreement between participants in the semantic clustering task, this is unsurprising. Although the analysis focused on maximal scores obtained per entity, if participants do not agree on groupings, then the means which are statistically compared are likely to mask a significant amount of variance. We now turn to the analysis by participants.

Results by participant The difference between target and baselines in agreement across participants was significant both for spatial ($t = 16.6$, $p < .01$) and semantic ($t = 5.759$, $t < .01$) domain types. This corroborates the earlier conclusion: once participant variation is controlled for by including it in the statistical model, the differences between target and baseline show up as reliable across the board. A univariate ANOVA corroborates the results, showing no significant main effect of domain type ($F < 1$, ns), but a highly significant main effect of algorithm ($F = 233.5$, $p < .01$) and a significant interaction ($F = 44.3$, $p < .01$).

Summary The results of the evaluation are encouraging, showing high agreement between the output of the algorithm and the output that was judged by humans as most appropriate. They also suggest framework of §4 corresponds to human intuitions better than the baselines tested here. However, these results should be interpreted with caution in the case of semantic clustering, where there was significant variability in human agreement. With respect to spatial clustering, one outstanding problem is that of reciprocal pairs which are too distant from eachother to form a perceptually well-formed cluster. We are extending the empirical study to new domains involving such cases, in order to infer from the human data a threshold on pairwise distance between entities, beyond which they are not clustered.

6 Conclusions and future work

This paper attempted to achieve a dual goal. First, we highlighted a number of scenarios in which the performance of a GRE algorithm can be enhanced by an initial step which identifies clusters of entities or properties. Second, we described an algorithm which takes as input a set of objects and returns a set of clusters based on a calculation of their *perceived proximity*. The definition of perceived proximity seeks to take into account some of the principles of human perceptual and conceptual organisation.

In current work, the algorithm is being applied to

two problems in GRE, namely, the generation of spatial references involving collective predicates (e.g. *gathered*), and the identification of the available perspectives or conceptual covers, under which referents may be described.

References

M. Aloni. 2002. Questions under cover. In D. Barker-Plummer, D. Beaver, J. van Benthem, and P. Scotto de Luzio, editors, *Words, Proofs, and Diagrams*. CSLI.

Anja Arts. 2004. *Overspecification in Instructive Texts*. Ph.D. thesis, Univiersity of Tilburg.

Robert Dale and Ehud Reiter. 1995. Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(8):233–263.

Robert Dale. 1989. Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, ACL-89*.

C. Eschenbach, C. Habel, M. Herweg, and K. Rehkamper. 1989. Remarks on plural anaphora. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics, EACL-89*.

K. Funakoshi, S. Watanabe, N. Kuriyama, and T. Tokunaga. 2004. Generating referring expressions using perceptual groups. In *Proceedings of the 3rd International Conference on Natural Language Generation, INLG-04*.

A. Gatt and K. van Deemter. 2005a. Semantic similarity and the generation of referring expressions: A first report. In *Proceedings of the 6th International Workshop on Computational Semantics, IWCS-6*.

A. Gatt and K. Van Deemter. 2005b. Towards a psycholinguistically-motivated algorithm for referring to sets: The role of semantic similarity. Technical report, TUNA Project, University of Aberdeen.

H.P. Grice. 1975. Logic and conversation. In P. Cole and J.L. Morgan, editors, *Syntax and Semantics: Speech Acts.*, volume III. Academic Press.

P. Jordan and M. Walker. 2000. Learning attribute selections for non-pronominal expressions. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, ACL-00*.

B. Kaup, S. Kelter, and C. Habel. 2002. Representing referents of plural expressions and resolving plural anaphors. *Language and Cognitive Processes*, 17(4):405–450.

A. Kilgarriff and D. Tugwell. 2001. Word sketch: Extraction and display of significant collocations for lexicography. In *Proceedings of the Collocations Workshop in Association with ACL-2001*.

S. Koh and C. Clifton. 2002. Resolution of the antecedent of a plural pronoun: Ontological categories and predicate symmetry. *Journal of Memory and Language*, 46:830–844.

E. Krahmer and M. Theune. 2002. Efficient context-sensitive generation of referring expressions. In Kees van Deemter and Rodger Kibble, editors, *Information Sharing: Reference and Presupposition in Language Generation and Interpretation.* Stanford: CSLI.

A. Kronfeld. 1989. Conversationally relevant descriptions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, ACL89*.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning.*

L. Moxey, A. J. Sanford, P. Sturt, and L. I Morrow. 2004. Constraints on the formation of plural reference objects: The influence of role, conjunction and type of description. *Journal of Memory and Language*, 51:346–364.

F. P. Prepaarata and M. A. Shamos. 1985. *Computational Geometry*. Springer.

E. Reiter. 1990. The computational complexity of avoiding conversational implicatures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, ACL-90.*

K. R. Thorisson. 1994. Simulated perceptual grouping: An application to human-computer interaction. In *Proceedings of the 16th Annual Conference of the Cognitive Science Society.*

A. Treisman. 1982. Perceptual grouping and attention in visual search for features and objects. *Journal of Experimental Psychology: Human Perception and Performance*, 8(2):194–214.

A. Tversky. 1977. Features of similarity. *Psychological Review*, 84(4):327–352.

K. van Deemter. 2000. Generating vague descriptions. In *Proceedings of the First International Conference on Natural Language Generation, INLG-00.*

Kees van Deemter. 2002. Generating referring expressions: Boolean extensions of the incremental algorithm. *Computational Linguistics*, 28(1):37–52.

M. Wertheimer. 1938. Laws of organization in perceptual forms. In W. Ellis, editor, *A Source Book of Gestalt Psychology.* Routledge & Kegan Paul.