Generating referring expressions

The basic procedure for identifying objects

Extension - relations between objects

Extension - sets of objects and boolean expressions

THE TASK

Complementary procedures for two alternatives

A pronominal reference

A noun phrase

potentially with coordinated expressions, and embedded relative clause

Generating a pronominal reference

Testing the uniqueness of grammatical features in context (last sentence)

Cautious strategies applied, no focus preferences and world knowledge impact

Generating specifications for a noun phrase

Mostly bulding a semantic specification independent of surface expersssions

Currently a hot topic in the field

THE BASIC TASK – IDENTIFICATION
GENERATING REFERRING EXPRESSIONS

Given

A set of objects, described in terms of entries in a knowledge base

Goal specification

A referring expression that identifies the intended referent(s) most naturally

Search strategy

Incrementally building referring expressions and testing their suitability

TERMINOLOGY

Intended referent

the entity to be described/ to be identified uniquely

Descriptor

an attribute or a relation applicable to an entity

Distinguishing description

a description only appying to the intended referent

Context set

the entities in the current focus of attention

Contrast set (potential distractors)

the entities in the context set other than the intended referent

Discriminatory power

degree of discrimination achievable by a descriptor

GENERATING REFERRING EXPRESSIONS - IN DETAIL

The task

Given a set of objects, descibed in terms of entries in a knowledge base

Build a referring expression that identifies the intended referent(s) naturally

By incrementally building referring expressions and test their suitability

Techniques applied

Solution in terms of compositions of elements of the knowledge base

Expression that is adequate and efficient (both factors need interpretation)

Depth-first, breadth-first, best-first, with iterative combinations

A FIRST ALGORITHM – FULL BREVITY (Dale 1989)

Functionality

Incrementally computes combinations of properties with increasing length

Alternative: Initial goal state chosen, improved by leaving out descriptors

Search strategy

Essentially breadth-first, cost (implicitly) not considered

Assessment

Finds optimal solution, computationally expensive

A POINT OF CRITIQUE

Evidence by psychological experiments

- **Humans produce “unnecessary” modifiers [Levelt 1989]**

objects	**x_1: bird, white**
	x_2: cup, white
	x_3: cup, black
	(often) “white bird” instead of “bird”

- **Humans produce expressions incrementally [Pechmann 1989]**
- **Properties are recognizable with varying speed (color better than shape)**
- **Situation-independent preference strategies**

THE INCREMENTAL ALGORITHM (Dale, Reiter 1995)

Functionality

Incrementally computes adds descriptors that have some discriminatory power

Ordering of descriptors according to domain-specific preferences

Search strategy

Pure depth-first, cost (implicitly) considered potentially high

Assessment

Finds reasonable, not always optimal solution, computationally efficient

A NON-OPTIMAL EXAMPLE

Goal

Identifiy cup$_1$

Context set

cup$_1$:	**<size,big>,**	**<color, red>,**	**<material,plastic>**
cup$_2$:	**<size,small>,**	**<color, red>,**	**<material,plastic>**
cup$_3$:	**<size,small>,**	**<color, red>,**	**<material,paper>**
cup$_4$:	**<size,middle>,**	**<color, red>,**	**<material,paper>**
cup$_5$:	**<size,big>,**	**<color, green>,**	**<material,paper>**
cup$_6$:	**<size,big>,**	**<color, blue>,**	**<material,paper>**
cup$_7$:	**<size,big>,**	**<color, blue>,**	**<material,plastic>**

Search result

<material,plastic> first chosen, but minimal description is “the big red cup”

DIFFERENT INTERPRETATIONS OF EFFICIENCY

Interpretation	*Complexity*
Full Brevity [Dale 1989]	$\approx n_a n_l$
Greedy Heuristic [Dale 1989]	$\approx n_a n_d n_l$
Local Brevity [Reiter 1990]	$\approx n_a n_d n_l$
Incremental Algorithm [Dale,Reiter 1991]	$\approx n_d n_l$

n_a … **number of descriptors applicable to the intended referent**

n_d … **number of potential distractors**

n_l … **number of attributes in the generated referring expression**

EXTENSION 1 – RELATIONS (Dale, Haddock 1991)

Functionality

Descriptors can also express relations to other objects

Identification task may be handed over to a related object

Search strategy

Originally pure depth-first

Assessment

Computationally efficient, but solution quality may be critical

PROBLEMS WITH RELATIONS (1)
THE ROLE OF KNOWLEDGE REPRESENTATION

Influence of knowledge representation 1 - involved situation

Discriminatory power of some descriptors "delayed"

Example: every object may be near to some other,

then "near-to" is not selected as a descriptor,

even though the description of the nearby object

may yield a considerable contrast set reduction

Influence of knowledge representation 2 - detailed modeling

Attributes may be modeled as relations to express details about the values

Example: color represented as a relation, to express color properties

the effect is the same as above, increases frequency

PROBLEMS WITH RELATIONS (2)
THE ROLE OF THE SEARCH STRATEGY

Consequences of the search strategy

Pure depth-first may yield unintuitive expressions (nested embeddings)

Recursion of algorithm to related objects needs modification:

no repetition of descriptors already used

identification of the original referent is of relevance only

Modifications of the search strategy

Depth-first combined with breadth-first - further descriptors of original referent

Priority lists of the original and all local referents combined

PROBLEMS WITH RELATIONS (3)
THE ROLE OF THE LINGUISTIC CONTEXT

Embedding of the descriptor selection process in the overall generation process

- **Descriptors accumulated in a algebraic expression**
- **Consequences for surface expressions not taken into account**

Potential remedies

- **Anticipate possible surface realizations**
- **Check whether a combination of realization alternatives is possible**
- **A specific problem with embeddings (relative clauses):**
  - **Anticipating potential scoping problems**

CONTROLING THE FORM OF SURFACE EXPRESSIONS

Techniques

Associate descriptors with surface positions

Limit place holders for each position (for coordinations, exclusion descriptions)

Search algorithm avoids expansions if limits would be exceeded

Example

surface position	*type*	*color*	*location*	*size*	*age*
head noun	•				
prenominal modifier		•	•	•	•
postnominal modifier			•		
relative clause	•	•	•	•	•

EXTENSION 2 – SETS OF OBJECTS (van Deemter 2000)

Functionality

Breadth-first within iterative deepening

***Boolean combinations* of attributes in addition to single attributes**

Increasingly complex combinations considered

(single attributes, combinations of two attributes, …)

Search strategy

Breadth-first within iterative deepening

Assessment

Computationally efficient, but solution quality may be very low

***Strong commitment* – A priori inclusion of structurally simpler combinations**

DEFICIT – AMBIGUITY (see Gardent 2002)

An example scenario

descriptors/objects	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
white	•	•	•	•	•	•	•	•	•	•	
dog		•						•	•	•	
cow			•	•	•	•	•				
big								•	•	•	
small						•	•				
medium-sized				•	•						
pitbul										•	
poodle									•		
holstein						•					
jersey					•						

DEFICIT – AMBIGUITY (cont'd)

x_5, x_6, x_9 and x_{10} are the intended referents

Attributes selected: 1) white (excluding x_{11}),

Then several possibilities, e.g.:

2) big ∨ cow (excluding x_1 and x_2)

3) Holstein ∨ ¬small (excluding x_7)

4) Jersey ∨ ¬medium (excluding x_4)

x_3 and x_8 still not excluded

“the white things that are big or a cow, a Holstein or not small, and a Jersey or not medium” instead of

“the pitbul, the poodle, the Holstein, and the Jersey”

by Gardent's complete constraint-based search (1,4 sec)

EXHAUSTIVE SEARCH WITH A CONSTRAINT SYSTEM (Gardent 2002)

Problem description

Identifying *a set* of referents (*S*) with *one* expression *L*

Accumulating descriptors, including *boolean combinations*

Problem modeling (truth (1+2) and contribution (3))

1. All properties in *L* $\supseteq$ all properties applicable to *S*

2. All negative properties in *L* $\supseteq$ all properties not applicable to *S*

3. For all distractors *C* of *S*: properties of *S* other than those of *C* > 0 or
non-properties of *S* but properties of *C* > 0

Extensions for disjunctions

A disjunction is a distinguishing description for a set of individuals *S*
if there is an element in the disjunction identifying covering subsets of *S*

SEARCHING WITH A CONSTRAINT SYSTEM

Distribution strategy (how to assign values to variables)

Case distinctions over cardinality of *L*, starting with minimal value

Algorithm stops once a solution is found

Implementation and results

Concurrent programming language Oz (PSE, Saarland University, 1998)

Supports set variables ranging over finite sets of integers

Runtime example: “the poodle, the jersey, the pitbul, and the poodle”

(10 objects, 10 descriptors, sparsely attributed) 1,4 sec

BEST-FIRST SEARCH (A*)

Properties

Homogenous evaluation of problem states required

Concept of optimal path costs: $f^*(n) = g^*(n) + h^*(n)$

Heuristic estimates of optimal path costs: $f(n) = g(n) + h(n)$

- **$g(n)$ … minimal path costs found to current state**
- **$h(n)$ … estimated path costs from current state to goal state**

Node associated with best heuristic score is expanded next

Theorem: If $h(n) \leq h^*(n)\ \forall\ n$, A* is *admissible* (finds optimal solution)

Use

Machine translation

Specific subprocesses in NLP

BEST-FIRST TECHNIQUE (Horacek 2003)
BEST-FIRST VERSUS INCREMENTAL SEARCHING

Description expansion

All **intermediate results can be expanded further**

Only the full expression in the incremental algorithm

Expansion point determination

Complexity of partial descriptions built so far

Number of potential distractors still to be excluded

Complexity of descriptor combinations still unused at specific state

BEST-FIRST SEARCHING

Method

Adding descriptors to one of the partial descriptions generated so far

Expansion according to complexity of partial descriptions and distractors excluded

***Efficiency measures – cut-offs* (assuming *conflation* is not possible)**

***Value* cut-off (global) – if a solution has been found, in an A^*-like fashion**

***Dominance* cut-off (local) – a sibling node, that is not superior in any aspect**

***Complexity* cut-off (individual) – description considered too complex**

Assessment

No redundancy, reasonable efficiency

SEARCH OPTIMIZATIONS - CUT-OFFS

Dominance Cut-off

Applicable to – Sibling nodes with

- **partial descriptions excluding the same potential distractors**
- **the same set of descriptors available**

The node with partial description evaluated worse is closed

Value Cut-off

If a solution is found, its score is compared to each node

If its partial score plus the optimistic estimate is below this score

Then this node is closed

ASSUMPTIONS

Value Cut-off

Descriptors map one-to-one onto surface expression components

Modification of simple counting possible (but must still be monotone)

Dominance Cut-off

Compositionality of expressions

Complexity Cut-off

Complex expressions impractical, task modification required

Partitioning the identification task, focus narrowing, then identification

Expression Cut-off

“Mixed” disjunctions impractical, task modification required

Partitioning the set of intended referents for separate identification

TAXONOMIC REASONING TO DETECT REDUNDANCIES

Generation of non-redundant boolean combinations

Critical part, very time-consuming

Burden partitioned between off-line taxonomic reasoning and dynamic generation

Taxonomic reasoning

***implies* (p,q) if *specializes*(p,q) holds**

***implies* (p,¬q) if *incompatible*(p,q) holds**

***implies* (¬p,q) if *opposite*(p,q) holds**

***implies* (¬p,¬q) if *generalizes*(p,q) holds**

Redundance-free generation of descriptor combinations evaluates

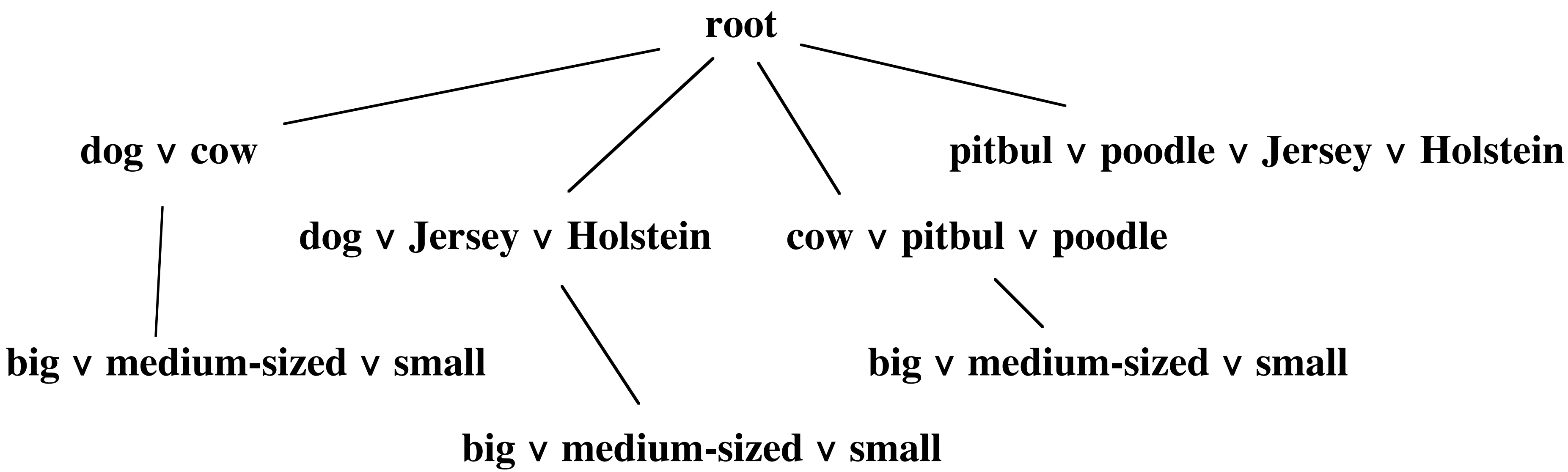
***subsumes*(p,q)**	**≡ *implies*(q,p)**	***p* not considered if *q* is**
***redundant*(p,q)**	**≡ (*subsumes*(q,p) ∨ *subsumes*(q,p))**	**at most one of *p*, *q* considered**

GENERATING DESCRIPTOR COMBINATIONS

1	***Nextprop*** **← <next combination subsuming target objects>**	**(*with given complexity*)**
	if *Nextprop* = nil then goto Step 2	**(*expressiveness test*)**
	if redundant(*p*,*q*) for any *p*,*q* ∈ *Nextprop* then goto Step 1	**(*redundancy test*)**
	if <*Nextprop* subsumes all distractors>	**(*discriminatory test*)**
	then goto Step 1	
	if <*Nextprop* subsumes fewer distractors than some sibling node>	
	then goto Step 1	**(*Dominance cut-off*)**
	return *Nextprop*	**(*solution found*)**
2	**if (Score(*Description*(*Best-Node*)) + Score(*Nextprop*)) ≥ *Complexity-limit***	
	then return nil	**(*Complexity cut-off*)**
	Nextprop **← Increment-size(*Nextprop*), goto Step 1**	**(*increasing complexity*)**

RESULTS

The example scenario



400 msec

(3,5x faster than Gardent)

INCREASED REPERTOIRE OF EXPRESSIVENESS

An example scenario

descriptors/*objects*	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
vehicle		•	•	•	•	•	•	•	•	•	•	•	•
car				•	•	•	•			•	•	•	•
sportscar						•	•					•	•
truck		•	•					•	•				
blue			•							•			•
red					•		•	•	•		•	•	
white		•		•		•							
center					•	•				•		•	
left			•						•		•		•
right		•		•			•	•					
big		•	•	•							•	•	•
small					•	•	•	•	•	•			
new		•			•	•			•		•		•
old			•	•			•	•		•		•	

INCREASED REPERTOIRE OF EXPRESSIVENESS

Composing descriptions of subsets

“the sportscars that are not red and the small trucks”

Identifying x_5, x_7, x_8, and x_{12} in two components, as opposed to

“the vehicles that are a sportscar or small are either a truck or not red”

An involved one-shot identification

Exclusion descriptions – describuing distractors rather than intended referents

“the vehicles on the right, but not the red truck”

Identifying x_1, x_3, and x_6 by explicitly excluding x_7

Sequence of increasingly restricting descriptions

“One of the trucks and the sportcars, all not white. The trucks stand in the center.”

Identifying x_6, x_7, x_{11}, and x_{12} in two stages

METHODS FOR ENHANCING EXPRESSIVENESS

Linguistically motivtaed preferences **(optional)**

Disjunctions of categories and attributes excluded ("car or red")

Controling suitability through complexity limitations

Associating descriptors with surface positions they can take

Generating a description with specified limitations about these positions

Generating sequences of such descriptions

Recasting descriptions

Transforming complex descriptions by applying distributivity

Opportunistically switching to describing distractors (at most once)

(see [Horacek 2004] for details)

RECASTING DESCRIPTIONS

Techniques

Partitioning a description according to descriptors and referents

Simplifications by eliminating non-existing combinations

Example

$\{x_5, x_7, x_8, x_{12}\}$ identified by (sportscar ∨ small) ∧ (truck ∨ ¬red)

3 possible partitionings, according to subexpression chosen and objects it covers

1. (sportscar ∧ (truck ∨ ¬red)) ∨ (small ∧ (truck ∨ ¬red)) for $\{x_{12}\}, \{x_5, x_7, x_8\}$

2. (sportscar ∧ (truck ∨ ¬red)) ∨ (small ∧ (truck ∨ ¬red)) for $\{x_5, x_{12}\}, \{x_7, x_8\}$

3. (truck ∧ (sportscar ∨ small)) ∨ (¬red ∧ (sportscar ∨ small))for $\{x_7, x_8\}, \{x_5, x_{12}\}$

2. and 3. (not 1.) can be simplified to (truck ∧ small) ∨ (¬red ∧ sportscar)

SWITCHING TO DESCRIPTIONS OF DISTRACTORS

Method

“Dual” task – identifying distractors rather than intended referents

Intended referents and distractors locally swaped

Applied at most once in a search branch

Criteria

Identification assessed simpler than direct identification of intended referents

Based on

- **the number of objects to be identified and**
- **the complexity of the next descriptor combination available**
- **effort to introduce exclusion phrase “ … , but” (specific criterion)**

AN EXAMPLE

***Intended referents*: {x_1 ,x_3 ,x_6}**

***Surface form restrictions*: head noun, pre- and post-nominal modifier, at most one of them in a conjoined expression, and a relative clause or a “but”-clause**

***First descriptor chosen*: 'right'**

***Distractors still to be excluded*: x_7**

***Next descriptor chosen*: 'car' ∨ 'white'**

***Partitioning (no coordination in relative clause)*: 'car' ∧ 'right' ∨ 'white' ∧ 'right'**

***Alternative* for 'car' ∨ 'white' – describing distractor x_7, with: 'truck'**

***Next descriptor chosen*: 'red', yielding 'right' ∧ ¬('truck' ∧ 'red')**

***Solutions*: “the vehicles on the right, but not the red truck”**

“the cars and the white vehicle, both on the right”

RESULTS IN TESTING EFFICIENCY MEASURES

Effects of the linguistically motivated restrictions

Effectiveness of the cut-off techniques

Behavior in scaling up for larger examples

Test cases

Subsets of 2, 3, 4 cars out of x_1 to x_6 (50 cases)

One version with all properties, one without size and age

GAIN BY LINGUISTIC PREFERENCES

	with *linguistic preferences*	*without* *linguistic preferences*
max. ***number of descriptors***	**5**	**5**
max. ***search tree size***	**9**	**20**
avg. ***search time (msec)***	**127.7**	**440.5**
max. ***search time (msec)***	**950**	**2590**

COMPARING CUT-OFF MEASURES

	cut-off measures			
	all	***value***	***dominance***	***complexity***
avg. ***search tree size***	**2.2**	**3.88**	**2.33**	**61.64**
max. ***search tree size***	**9**	**71**	**11**	**945**
avg. ***search time (msec)***	**127.7**	**168.1**	**595.0**	**1133.1**
max. ***search time (msec)***	**690**	**2320**	**4550**	**19210**

EXAMINING SCALABILITY

	Number of distractors			
	6	**9**	**12**	**25**
max. ***search tree size***	**9**	**16**	**61**	**907**
min. ***search time (msec)***	**10**	**10**	**30**	**120**
avg. ***search time (msec)***	**116**	**484**	**1120**	**24838**
max. ***search time (msec)***	**490**	**4100**	**6530**	**141200**

PROBLEMS WITH SETS OF OBJECTS

Complexity of expressions

Up to 8 descriptors for the scenario with 12 objects

Extreme example

"the cars which are not blue, are old or stand in the center, are new or stand on the right side, are big or not white, and are small or not red"

108110 msec, identifying x_3, x_4, and x_6 out of 25 vehicles

Measures

Other search methods (full computation, best-first)

Splitting the task into subgroups of intended referents

EMPIRICAL APPROACHES

Research questions

Learn about human preferences: attributes used, cooccurrences, minimality, …

Experimemntal setting - the TUNA corpus

Grid-based situation (3x5 cells) with a small set of entities (5-6), 1 intended referent

Two different sets of tests - furniture item, and people

Categories (chair), qualitative (bearded) and vague descriptors (large, old), location

Evidence

Noisy data – some expressions non-felicitous or ambiguous

Some attributes used frequently – category, salient properties (beard, glasses)

Some use of non-minimal descriptions, regularity hard to find

Some personal styles – intrinsic properties preferred vs. location preferred

TUNA CHALLENGE (1)

Setting

Corpus divided into training and test subcorpora (ca. 80/20%)

Expression preferred by human subject, abstracted into descriptors

Evaluation (only attribute selection)

Attribute sets A and B (machine produced vs. human "gold standard")

$$Dice(A,B) = \frac{2x\,|A \cap B|}{|A| + |B|}$$ **between 0 and 1, 1 means a perfect match**

$$MASI(A,B) = \delta \times \frac{|A \cap B|}{|A \cup B|} \qquad \partial = \begin{cases} 0 & \text{if } A \cap B = \emptyset \\ 1 & \text{if } A = B \\ \frac{2}{3} & \text{if } A \subset B \text{ or } B \subset A \\ \frac{1}{3} & \text{otherwise} \end{cases}$$ **monotonicity coefficient**

Coefficients computed for complete set of test corpus examples

String edit and BLEU scores used for end-to-end and realization competitions

TUNA CHALLENGE (2)

Some of the techniques used

Choosing most frequently used descriptors (+ value) according to setting

Type attribute always included, orthers only if they contribute to discrimination

Incremental algorithm applied

Either both location descriptors (x-, y-) or none

Nearest neighbor – most similar expression (Dice) of the same subject

Individuation – mimicking preferences of the specific subject in a trial

Results

Most fine-grained criteria and learning techniques prove benefcial

Individuation pays off

Best scores almost 0.9 (Dice) and almost 0.8 (Masi)

REFERENCES IN HIERARCHICAL DOMAINS (1)

Examples

Documents - Figure (in paragraph) in section …

Spatial areas - Room number/name in building …

Problems

Uniquely identifying descriptions may be difficult to find (e. g., room 1)

Extra attributes indicating hierarchical scope support easy identification

Lack of orientation **- addressee tries to identify descripition nearby**

Dead end **– adressee might try to identify the intended referent in a wrong scope**

Algorithmic modifications

Compromises between confidence (1) and conciseness (2)

1. Incrementally adding descriptors to obtain unique identifiability in wider scope

2. Only adding attributes that are needed for distinction in wider scopes

REFERENCES IN HIERARCHICAL DOMAINS (2)

Experimental settings

Confronting the subjects with a set of alternatives

Minimal and extended expressions

- **The green star is shown in 1. Part C of Section 2 or 2. part C**
- **The green star is shown in 1. Table 2 in Part B of Section 2 or 2. Table 2**

Hypotheses (summary)

1. In problematic situations, redundant expressions are preferred

2. In non-problematic situations, the full description is dispreferred

Results

1. confirmed - highly significant

2. not confirmed - as a trend, but considerable differences between subjects

VAGUENESS - THE ISSUE

One frequent manifestation of vagueness are gradables

"the large(st) mouse", "the (n) large(st) mice"

> **Base form implies some standard of the measures appearing in context**

Referential uses in sequences of utterances (no implications):

> **"the large mouse … " "dozens of mice …" "the large mouse … "**

Evaluatives (no inference about converses):

> **"Hans is taller than Fritz" => "Fritz is shorter than Hans"**
>
> **"Hans is smarter than Fritz" ≠> "Fritz is more stupid than Hans"**

Relative and absolute values:

> **"The short man" (Fritz, 2m, vs. Hans 2m 5cm) seems odd**

A further problem: are small differences observable?

VAGUENESS IN GENERATION (van Deemter)

Representation

Distinction between

- **measurable (internal) properties (e.g., "height = 10 cm", "width < 6 cm")**
- **gradables (to be used in "natural" expressions)**

Algorithm (sketch)

measurable properties mapped onto intervals (e.g., "size > 10 cm", "size < 6 cm")

applicable intervals used as descriptors

mapped onto expressions built out of gradables

most gradable properties dispreferred to most other descriptors

(in ordered preference list)

Surface generation incorporates pragmatic constraints

SOME FURTHER ISSUES IN GENERATING REFERRING EXPRESSIONS

Multimodal referring expressions

Effects of language and culture

Uncertainties about the recognition/knowledge of the addressee

Implicature of expressions

Guiding the focus of attention

Integration into the whole generation task (e.g., surface realization)