Referring Expression Generation: What Can We Learn From Human Data?

Robert Dale and Jette Viethen
rdale | jviethen@science.mq.edu.au

The Aims of This Talk

- To review the development of algorithms for referring expression generation
- To argue that existing algorithms are not a good starting point for modelling what people do
- To suggest a different way of looking at the referring expression generation task

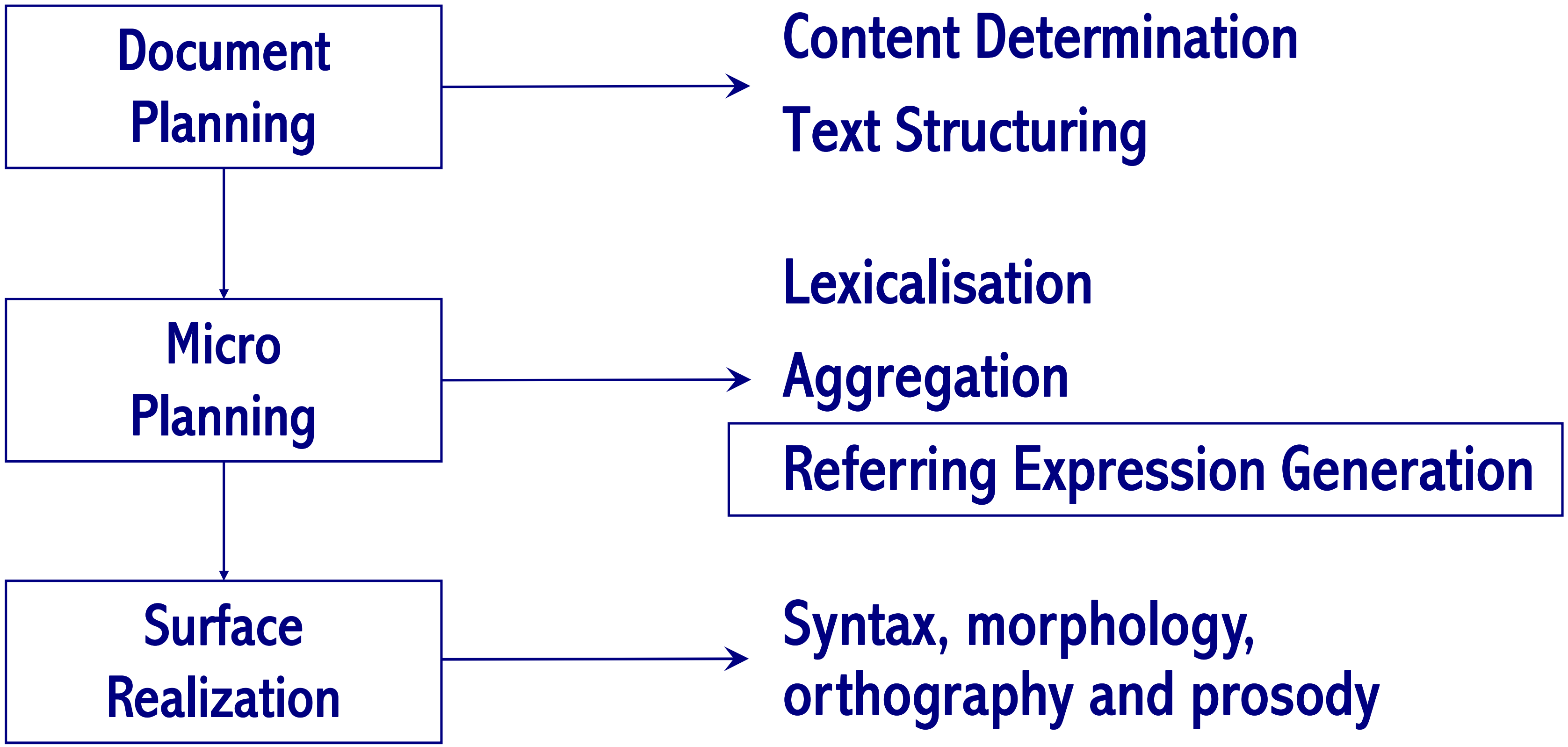
Outline

- Referring Expression Generation: The Current Paradigm
- What People Do
- A Different Paradigm: Attribute-Based Heuristics
- Where Next?

The Context: Natural Language Generation

- Natural Language Generation is concerned with generating linguistic material from some non-linguistic base
- Why is this important?
  - Applications:
    - any situation where it is not practical to construct the full range of required outputs ahead of time
  - Theory:
    - understanding what drives choice-making in language

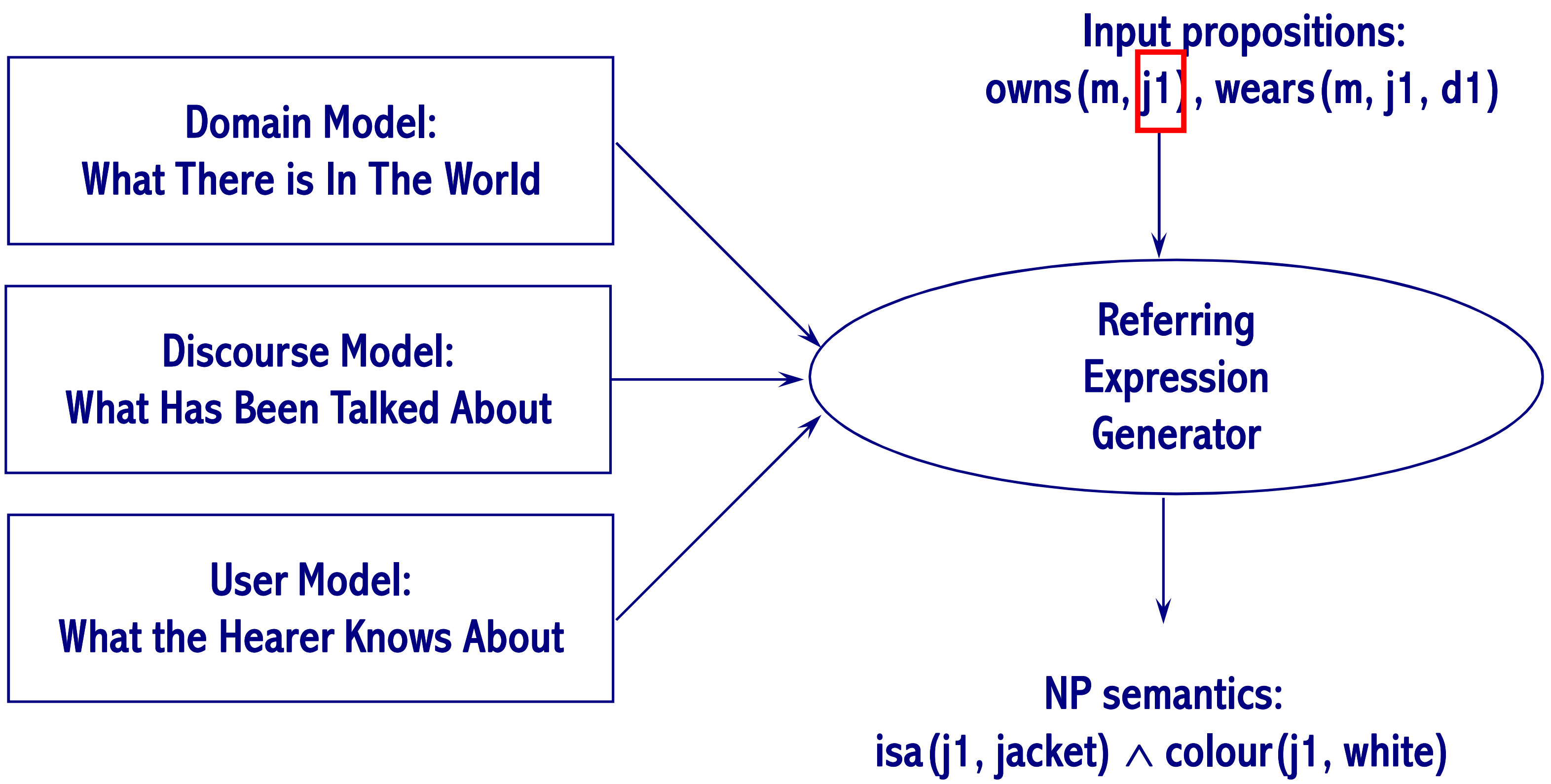
The Natural Language Generation Pipeline



What's Involved in Referring Expression Generation?

- The Task:
  - Given some entity we want to refer to, represented by some internal symbol, how do we go about deciding how to refer to it?
- Governed by neo-Gricean Principles:
  - Adequacy
  - Efficiency
  - Sensitivity

Referring Expression Generation



The Effect of Discourse Context on Reference

- Example 1:
  - owns(m, j1) → Matt owns a white jacket.
  - wears(m, j1, d) → He wears it on Sundays.

Different

- Example 2:
  - owns(m, [j1+c1]) → Matt owns a white jacket and a white coat.
  - wears(m, j1, d) → He wears the jacket on Sundays.

Same

- Example 3:
  - owns(m, [j1+j2]) →Matt owns a white jacket and a blue jacket.
  - wears(m, j1, d) → He wears the white one on Sundays.

The Consensus Problem Statement

- Given
  - an intended referent R
  - a contrast set C consisting of the potential distractor entities
  - knowledge of the properties of the entities
- ... find a set of properties true of R that, together, are not true of any entity in C.
- The result is a <u>distinguishing description</u> of R.

Computing Distinguishing Descriptions

Three steps which are repeated until a successful description has been constructed:

Start with a null description.

1. Check whether the description constructed so far is successful in picking out the intended referent from the context set. If so, quit.
2. If it's not sufficient, choose a property that will contribute to the description.
3. Extend the description with this property, and reduce the context set accordingly. Go to Step 1.

Computing Distinguishing Descriptions: The Greedy Algorithm

Initial Conditions:

$C_r = \langle$all entities$\rangle$; $P_r = \langle$all properties true of r$\rangle$; $L_r = \{\}$

1. Check Success

   if $|C_r| = 1$ then return L_r as a distinguishing description

   elseif $P_r = 0$ then return L_r as a non-dd

   else goto Step 2.

2. Choose Property

   for each $p_i \in P_r$ do: $C_{r_i} \leftarrow C_r \cap \{x \mid p_i(x)\}$

   Chosen property is p_j, where C_{r_j} is smallest set.

   goto Step 3.

3. Extend Description (wrt the chosen p_j)

   $L_r \leftarrow L_r \cup \{p_j\}$; $C_r \leftarrow C_{r_j}$; $P_r \leftarrow P_r - \{p_j\}$; goto Step 1. [Dale 1987]

Problems with This Algorithm

- The algorithm does not guarantee to find a minimal distinguishing description
- Some properties are more useful than other properties which have the same discriminatory power

A Response: The Incremental Algorithm

Initial Conditions:

- $C_r = \langle \text{all entities} \rangle$; $P = \langle \text{preferred attributes} \rangle$; $L_r = \{\}$

1. Check Success
   - if $|C_r| = 1$ then return L_r as a distinguishing description
   - elseif $P = 0$ then return L_r as a non-dd
   - else goto Step 2.
2. Evaluate Next Property
   - get next $p_i \in P$ such that userknows$(p_i(r))$
   - if $|\{x \in C_r \mid p_i(x)\}| < |C_r|$ then goto Step 3
   - else goto Step 2.
3. Extend Description (wrt the chosen p_j)
   - $L_r \leftarrow L_r \cup \{p_j\}$; $C_r \leftarrow C_{rj}$; goto Step 1.

[Reiter and Dale 1992]

Key Properties of the Incremental Algorithm

- Embodies a distinction between:
  - the way choices are made (domain independent)
  - the choices available (domain dependent)
- May generate redundant descriptions
  - This is seen as a good thing

Extensions to the Basic Algorithms: Relations

- What happens if you need to mention another entity in order to identify the intended referent?
  - ‘the dog next to the small cat’
- Extensions to incorporate relations:
  - constraint-based extension for relational properties [Dale and Haddock 1991]
  - referring to parts of hierarchically structured objects [Horacek 2006]

Extensions to the Basic Algorithms: Disjunction and Negation of Properties

- What happens if there are multiple entities instead of one?
  - ‘the two dogs’
  - ‘the dog and the cat’
- What happens if a distinguishing characteristic is that the intended referent lacks some property?
  - ‘the dog that isn’t a poodle’
- Extensions:
  - Sets [Stone 2000]
  - Negation and Disjunction [van Deemter 2002]:

The General Form of These Algorithms

Given an intended referent R, a set of distractors C, a set of properties L_R, and the set of properties D to use in a description:

let D = {}
repeat
    add a selected property $\in L_R$ to D
    recompute C given D
until C = {}

How Algorithms Differ: The Selection of Properties

- The Greedy Algorithm [Dale 1989]
  - Check all the properties, see which one has the greatest discriminatory power
- The Incremental Algorithm [Dale and Reiter 1995]
  - Take the next property from a predetermined list, provided it has some discriminatory power
- The Graph-Based Algorithm [Krahmer et al 2003]
  - Take the property that has the greatest weight/least cost

Why Is This Not a Good Model of What People Do?

1. People often produce redundant descriptions
2. People don't always produce distinguishing descriptions
3. Different people produce different descriptions in the same situation
4. The 'add a property, check how we're doing' model seems too computationally expensive to be plausible

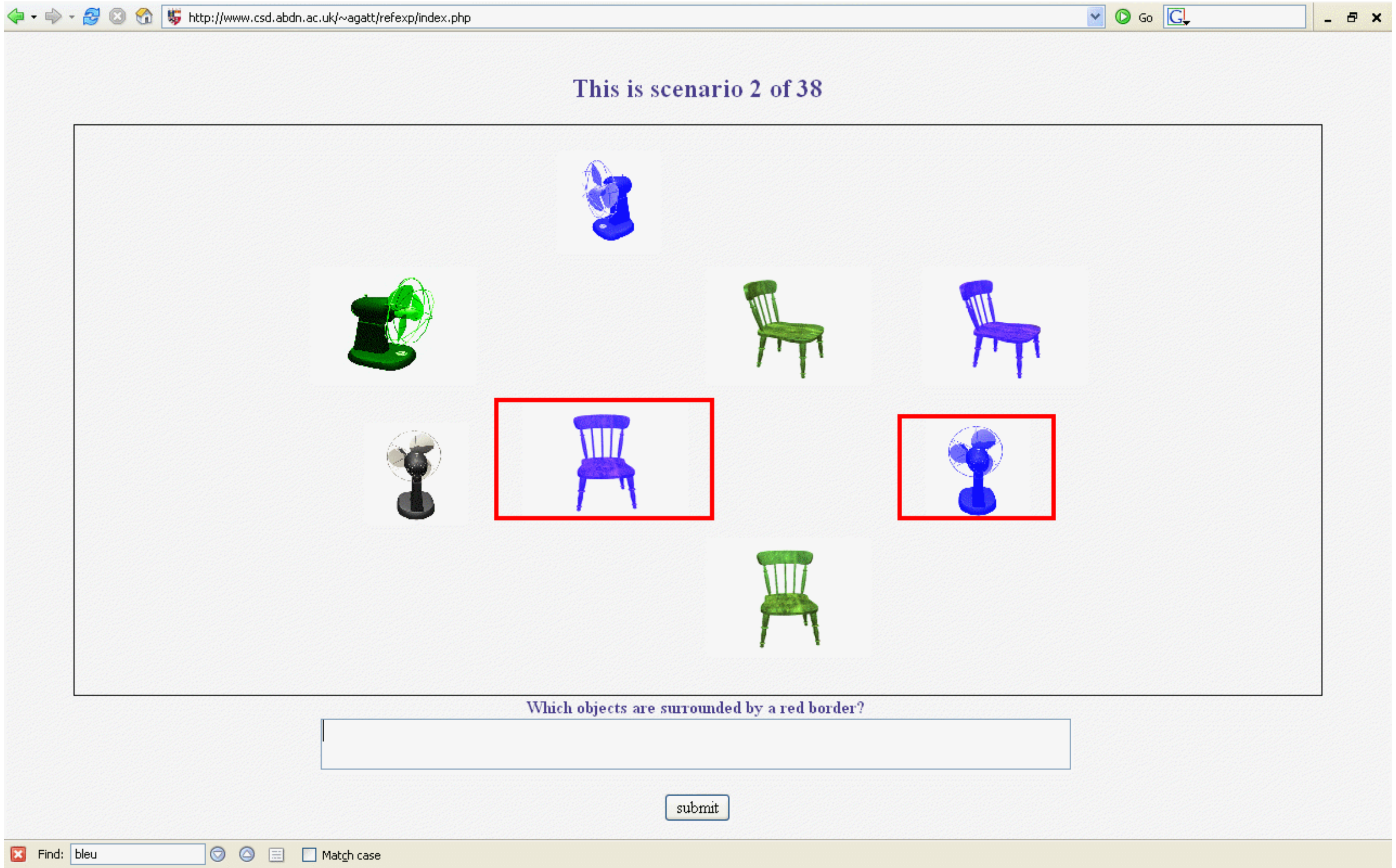
Outline

- Referring Expression Generation: The Current Paradigm
- What People Do
- A Different Paradigm: Attribute-Based Heuristics
- Where Next?

Human-Produced Data Sets

- The TUNA Corpus [van Deemter et al 2006]
  – 900 descriptions of furniture
  – 900 descriptions of people
- The GRE3D3 Corpus [Viethen and Dale 2008]
  – 630 descriptions of coloured blocks

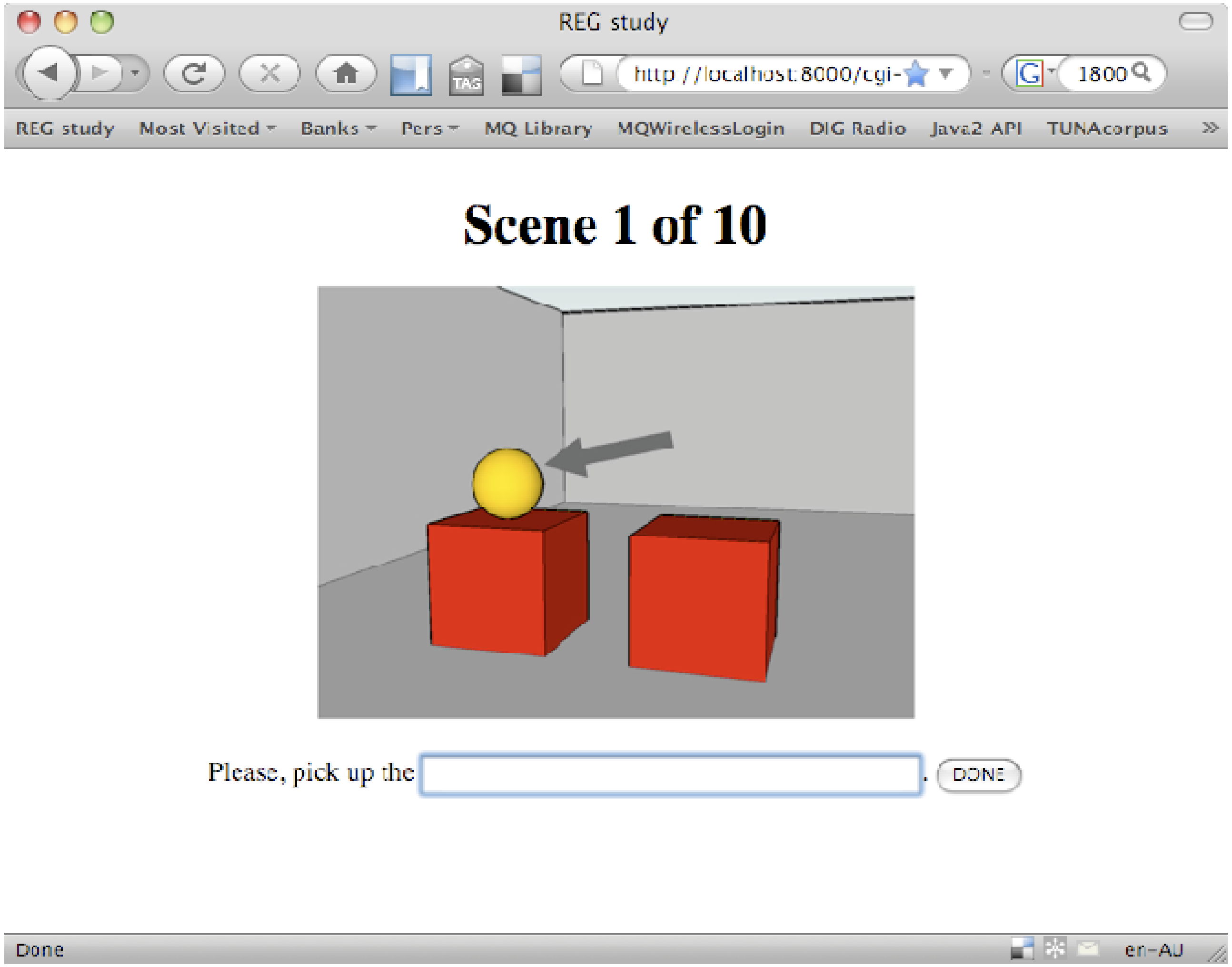
TUNA Furniture



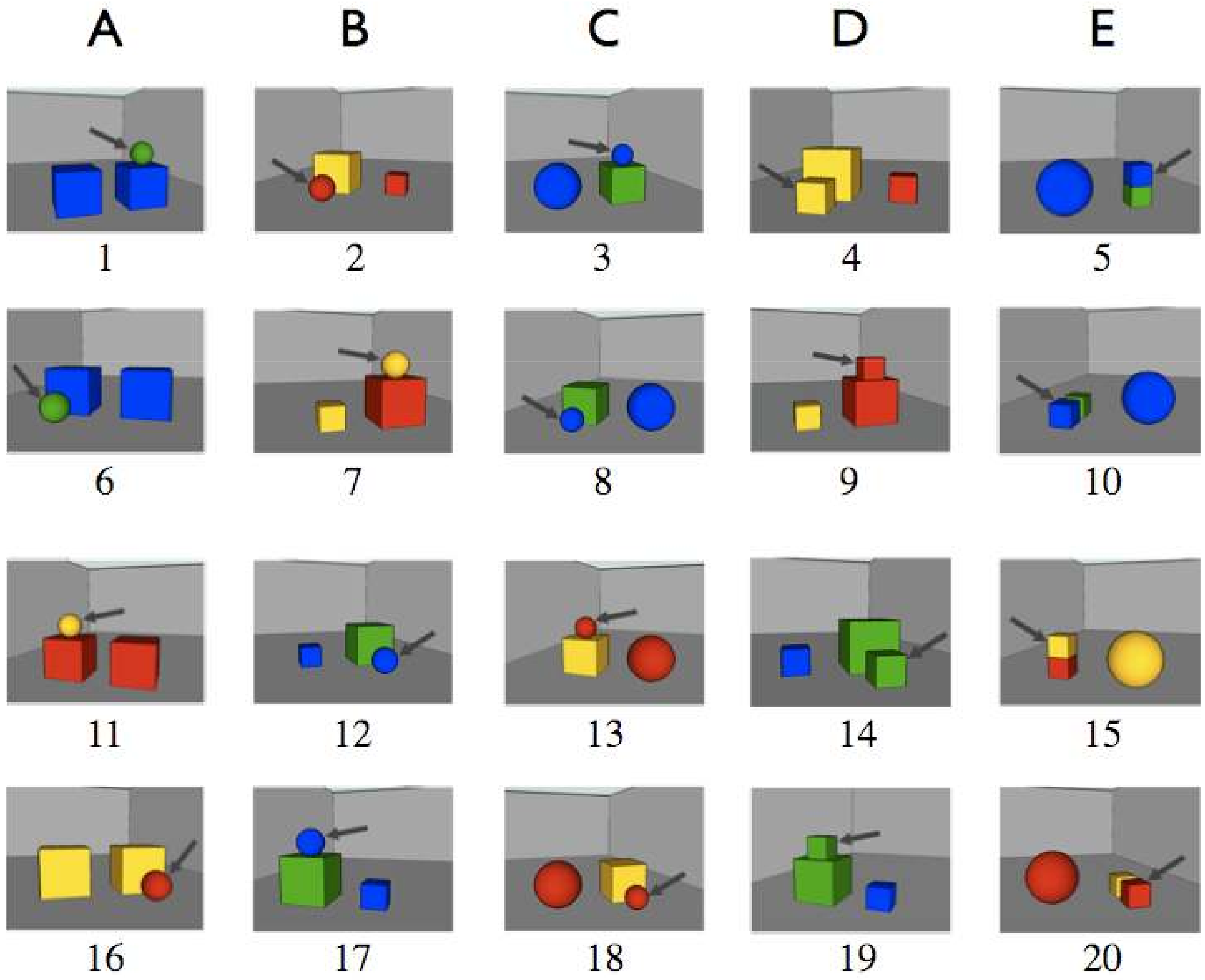
The GRE3D3 Corpus

- Research question: Do people use relations only when they are absolutely necessary?
- Materials: 20 different simple blocksworld scenes containing three objects, split into two trials; each subject sees 10 scenes
- Task: subject has to provide a distinguishing description in each scene for one of the objects; scenes constructed so that relations are never necessary
- Subjects: 74 participants recruited via the Internet

The Experimental Setup



The Stimulus Scenes



Data Filtering and Normalisation

- 74 participants:
  - One asked for data to be discarded; one reported as being colour blind; one used very long referring expressions referring to the onlooker; eight participants only used type in their descriptions
- Normalisation:
  - Spelling mistakes corrected; colour names and head nouns normalised; complex syntactic structures simplified

→ 623 scene descriptions

Relation Use

- Over a third (231 or 36.6%) of the descriptions use spatial relations
- 40 (63.5%) of the 63 participants used relations
- 23 (36.5%) of the participants never used relations
- 11 (over 25%) of the relation-using participants did so in all 10 referring expressions they delivered

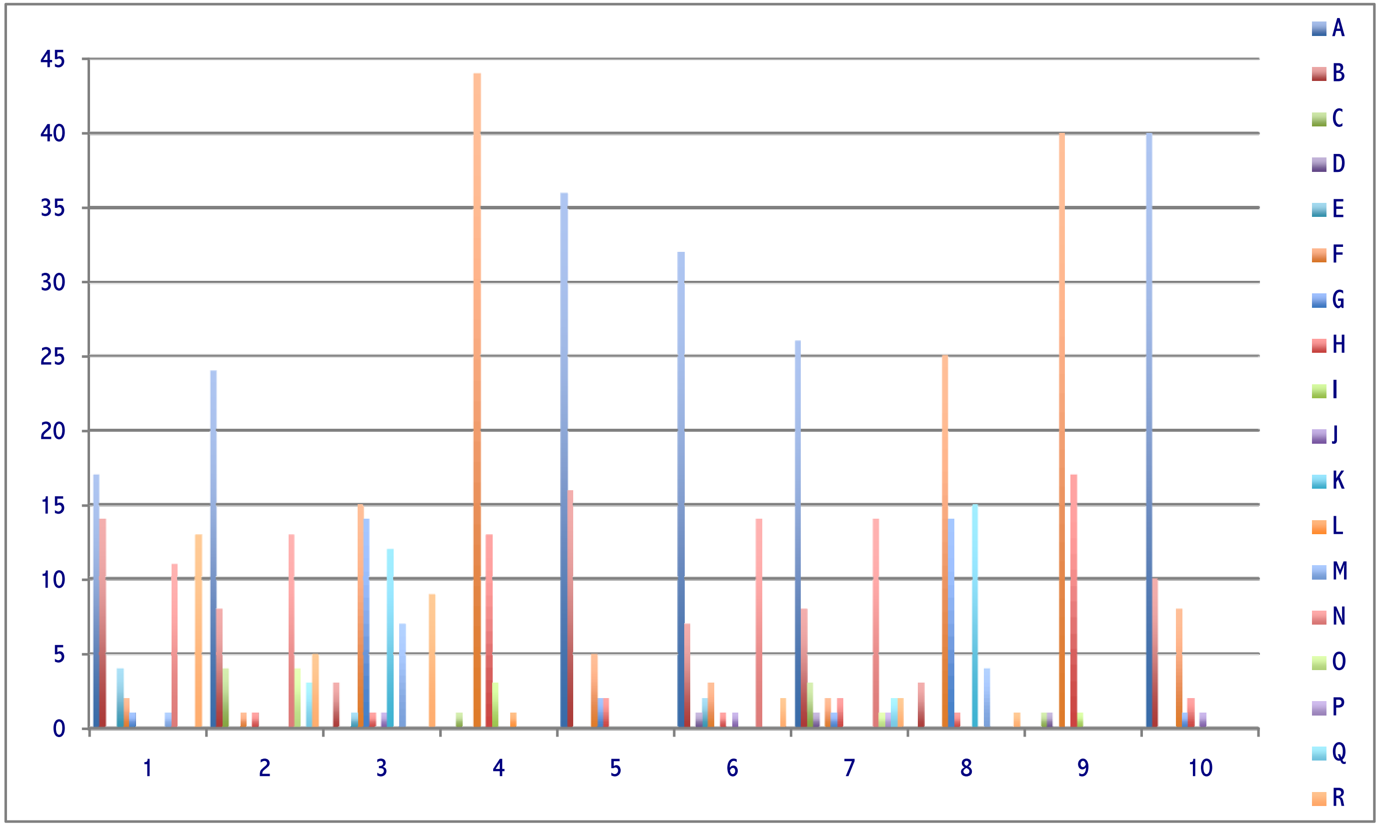
Interim Conclusions

- There are three kinds of people in the world – those who always use relations, those who never use relations, and those who sometimes do …
- There was a tendency for relations to be used less for later scenes: people learn that they are not necessary?
- But most importantly: people just do lots of different things

Description Patterns

Label	Pattern	Example
A	⟨tg_col, tg_type⟩	*the blue cube*
B	⟨tg_col, tg_type, rel, lm_col, lm_type⟩	*the blue cube in front of the red ball*
C	⟨tg_col, tg_type, rel, lm_size, lm_col, lm_type⟩	*the blue cube in front of the large red ball*
D	⟨tg_col, tg_type, rel, lm_size, lm_type⟩	*the blue cube in front of the large ball*
E	⟨tg_col, tg_type, rel, lm_type⟩	*the blue cube in front of the ball*
F	⟨tg_size, tg_col, tg_type⟩	*the large blue cube*
G	⟨tg_size, tg_col, tg_type, rel, lm_col, lm_type⟩	*the large blue cube in front of the red ball*
H	⟨tg_size, tg_col, tg_type, rel, lm_size, lm_col, lm_type⟩	*the large blue cube in front of the large red ball*
I	⟨tg_size, tg_col, tg_type, rel, lm_size, lm_type⟩	*the large blue cube in front of the large ball*
J	⟨tg_size, tg_col, tg_type, rel, lm_type⟩	*the large blue cube in front of the ball*
K	⟨tg_size, tg_type⟩	*the large cube*
L	⟨tg_size, tg_type, rel, lm_size, lm_type⟩	*the large cube in front of the large ball*
M	⟨tg_size, tg_type, rel, lm_type⟩	*the large cube in front of the ball*
N	⟨tg_type⟩	*the cube*
O	⟨tg_type, rel, lm_col, lm_type⟩	*the cube in front of the red ball*
P	⟨tg_type, rel, lm_size, lm_col, lm_type⟩	*the cube in front of the large red ball*
Q	⟨tg_type, rel, lm_size, lm_type⟩	*the cube in front of the large ball*
R	⟨tg_type, rel, lm_type⟩	*the cube in front of the ball*

Distribution of Patterns Across Scenes



Distribution of Patterns Across Scenes

Pattern		Scene #									
		1	2	3	4	5	6	7	8	9	10
A	tg_col, tg_type	17	24			36	32	26			40
B	tg_col, tg_type, rel, lm_col, lm_type	14	8	3		16	7	8	3		10
C	tg_col, tg_type, rel, lm_size, lm_col, lm_type		4		1			3		1	
D	tg_col, tg_type, rel, lm_size, lm_type						1	1		1	
E	tg_col, tg_type, rel, lm_type	4		1			2				
F	tg_size, tg_col, tg_type	2	1	15	44	5	3	2	25	40	8
G	tg_size, tg_col, tg_type, rel, lm_col, lm_type	1		14		2		1	14		1
H	tg_size, tg_col, tg_type, rel, lm_size, lm_col, lm_type		1	1	13	2	1	2	1	17	2
I	tg_size, tg_col, tg_type, rel, lm_size, lm_type				3					1	
J	tg_size, tg_col, tg_type, rel, lm_type			1			1				1
K	tg_size, tg_type			12					15		
L	tg_size, tg_type, rel, lm_size, lm_type				1						
M	tg_size, tg_type, rel, lm_type	1		7					4		
N	tg_type	11	13				14	14			
O	tg_type, rel, lm_col, lm_type		4					1			
P	tg_type, rel, lm_size, lm_col, lm_type							1			
Q	tg_type, rel, lm_size, lm_type		3					2			
R	tg_type, rel, lm_type	13	5	9			2	2	1		

Can We Use this Data to Learn How to Refer?

1. Identify relevant characteristics of scenes
2. See if these can be correlated with patterns via a machine learner

Characteristics of Scenes

Label	Attribute	Values
tg_type = lm_type	Target and Landmark share Type	TRUE, FALSE
tg_type = dr_type	Target and Distractor share Type	TRUE, FALSE
lm_type = dr_type	Landmark and Distractor share Type	TRUE, FALSE
tg_col = lm_col	Target and Landmark share Colour	TRUE, FALSE
tg_col = dr_col	Target and Distractor share Colour	TRUE, FALSE
lm_col = dr_col	Landmark and Distractor share Colour	TRUE, FALSE
tg_size = lm_size	Target and Landmark share Size	TRUE, FALSE
tg_size = dr_size	Target and Distractor share Size	TRUE, FALSE
lm_size = dr_size	Landmark and Distractor share Size	TRUE, FALSE
rel	Relation between Target and Landmark	on top of, in front of

Results

- Weka J48 pruned decision tree classifier
- Predicts actual form of reference in 48% of cases under 10-fold cross validation
- The rule learned:

```
if tg_type = dr_type
    then use pattern F (⟨tg_size, tg_col, tg_type⟩)
    else use pattern A (⟨ tg_col, tg_type⟩)
endif
```

Distribution of Patterns Across Scenes

Pattern	Scene # 1	2	3	4	5	6	7	8	9	10
A tg_col, tg_type	17	24			36	32	26			40
B tg_col, tg_type, rel, lm_col, lm_type	14	8	3		16	7	8	3		10
C tg_col, tg_type, rel, lm_size, lm_col, lm_type		4		1			3		1	
D tg_col, tg_type, rel, lm_size, lm_type						1	1		1	
E tg_col, tg_type, rel, lm_type	4		1			2				
F tg_size, tg_col, tg_type	2	1	15	44	5	3	2	25	40	8
G tg_size, tg_col, tg_type, rel, lm_col, lm_type	1		14		2		1	14		1
H tg_size, tg_col, tg_type, rel, lm_size, lm_col, lm_type		1	1	13	2	1	2	1	17	2
I tg_size, tg_col, tg_type, rel, lm_size, lm_type				3					1	
J tg_size, tg_col, tg_type, rel, lm_type			1			1				1
K tg_size, tg_type			12					15		
L tg_size, tg_type, rel, lm_size, lm_type				1						
M tg_size, tg_type, rel, lm_type	1		7					4		
N tg_type	11	13				14	14			
O tg_type, rel, lm_col, lm_type		4					1			
P tg_type, rel, lm_size, lm_col, lm_type							1			
Q tg_type, rel, lm_size, lm_type		3					2			
R tg_type, rel, lm_type	13	5	9			2	2	1		

What About Speaker Difference?

- As well as the characteristics of scenes, add participant ID as a feature
- Description pattern prediction increases to 57.62%
- So: it may be possible to learn individual differences from the data

Interim Conclusions

- We can learn a ‘correct answer’ for every scene
- We can’t explain the diversity in forms of reference

Outline

- Referring Expression Generation: The Current Paradigm
- What People Do
- A Different Paradigm: Attribute-Based Heuristics
- Where Next?

The Basic Idea

- People build different *descriptions* for the same intended referent in the same scene
- But maybe the decision processes around each specific *attribute* are less varied?

Learning the Presence or Absence of Individual Properties

Attribute to Include	Baseline (0-R)
Target Colour	78.33%
Target Size	57.46%
Relation	64.04%
Landmark Colour	74.80%
Landmark Size	88.92%

Heuristics for Colour Inclusion

- Always use colour [37 participants]
- If the target and the landmark are of the same type, use colour [all the rest]
- If the target and the landmark are not of the same type then:
  - Ignore colour [19 participants]
  - Use colour if target and distractor are the same size [4]
  - Use colour if target and distractor share size and the target is on top of the landmark [2]
  - Use colour if target and distractor share colour [1]

What Does This Mean?

- Everybody's different, but we often have some things in common:
  - A <u>speaker profile</u> consists of a collection of <u>attribute-specific heuristics</u>
  - Speaker profiles can vary significantly but be based on a set of commonly used attribute-specific heuristics
- The heuristics a particular speaker uses in a given situation may depend on a variety of contextual and personal-history factors

Speaker Profiles

#	tg_col	tg_size	tg_size	rel	lm_size
13	TgCol-T	TgSize-1	Rel-F	n/a	n/a
10	TgCol-T	TgSize-1	Rel-T	LmCol-T	LmSize-1
9	TgCol-1	TgSize-1	Rel-F	n/a	n/a
2	TgCol-3	TgSize-1	Rel-4	LmCol-F	LmSize-1
2	TgCol-T	TgSize-1	Rel-2	LmCol-T	LmSize-1
2	TgCol-1	TgSize-1	Rel-T	LmCol-1	LmSize-1

- TgCol-T = always include tg colour
- TgSize-1 = include tg size if tg and dr share type
- Rel-F = never use a relation

Implications for Algorithm Development

- Each property is different: reduction to a single metric of value (such as discriminatory power) is too simplistic
- Properties may be included independently of other properties
- An alternative to the ‘add one then check’ model:
  - A ‘read off the scene’ model: gestalt analysis of a scene results in several properties being chosen in parallel

Outline

- Referring Expression Generation: The Current Paradigm
- What People Do
- A Different Paradigm: Attribute-Based Heuristics
- Where Next?

Is This The Whole Story?

- No. Sometimes we <u>do</u> reflect on the referring expression constructed so far, and add more:
  - Uhm, I'm gonna transfer to the phone on the table by the red chair . . . [points in the direction of the phone] the . . . the red chair, against the wall, uh the little table, with the lamp on it, the lamp that we moved from the corner? . . . the black phone, not the brown phone . . .

[Lucy from Twin Peaks]

New Questions

- What properties of a scene just ‘jump out’?
- How do we decide if the first cut is good enough?
- What kinds of reasoning are involved in determining what else is needed in a referring expression?
- How and when do more reflective reasoning processes kick in?
- How are speaker profiles modified dynamically through alignment and learned success?

Conclusions

- Existing algorithms, based on a cycle of 'add a carefully-considered property then check how we're doing', don't acknowledge 'bounded rationality'
- Hypothesis: different speakers use different heuristics for property inclusion in different circumstances, based on individual history and other factors
- To build a psycholinguistically plausible model, we need to explore what conditions the use of specific properties