The Generation of Referring Expressions: Where We've Been, How We Got Here, and Where We're Going

Robert Dale
Robert.Dale@mq.edu.au

The Aims of This Talk

- To outline what referring expression generation is about
- To characterise the current state of the art and developments in the field
- To outline an agenda for future work in the area

Outline

- The Context: Natural Language Generation
- The Story So Far: Algorithm Development to Empiricism
- Challenges for the Future

The Context

- 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

Natural Language Generation Applications

- Generating text from large data sets:
  - Weather reports, stock market reports
- Information personalisation:
  - Tailored web pages that take account of what you know
- Context-sensitive generation:
  - Dynamic utterance construction in dialog systems
- Multilingual generation:
  - Multiple languages from a common knowledge source

NL Understanding vs NL Generation

- The view from Natural Language Understanding:
  - Deriving meaning from text means throwing away or ignoring irrelevant detail
- The view from Natural Language Generation:
  - Very few, if any, surface variations are meaningless; we need to explain their function if we are to understand them properly

Mapping Between Representations: NLU

$[e,x,y]\mathrm{kissing}(e) \wedge \mathrm{past}(e) \wedge$
$\mathrm{name}(x,\mathrm{John}) \wedge \mathrm{name}(y,\mathrm{Mary}) \wedge$
$\mathrm{agent}(e,x) \wedge \mathrm{patient}(e,y)$

John kissed Mary.

Mary was kissed by John.

Mapping Between Representations: NLG

[e,x,y]kissing(e) ∧ past(e) ∧
name(x,John) ∧ name(y,Mary) ∧
agent(e,x) ∧ patient(e,y)

... ∧ focus(x)

... ∧ focus(y)

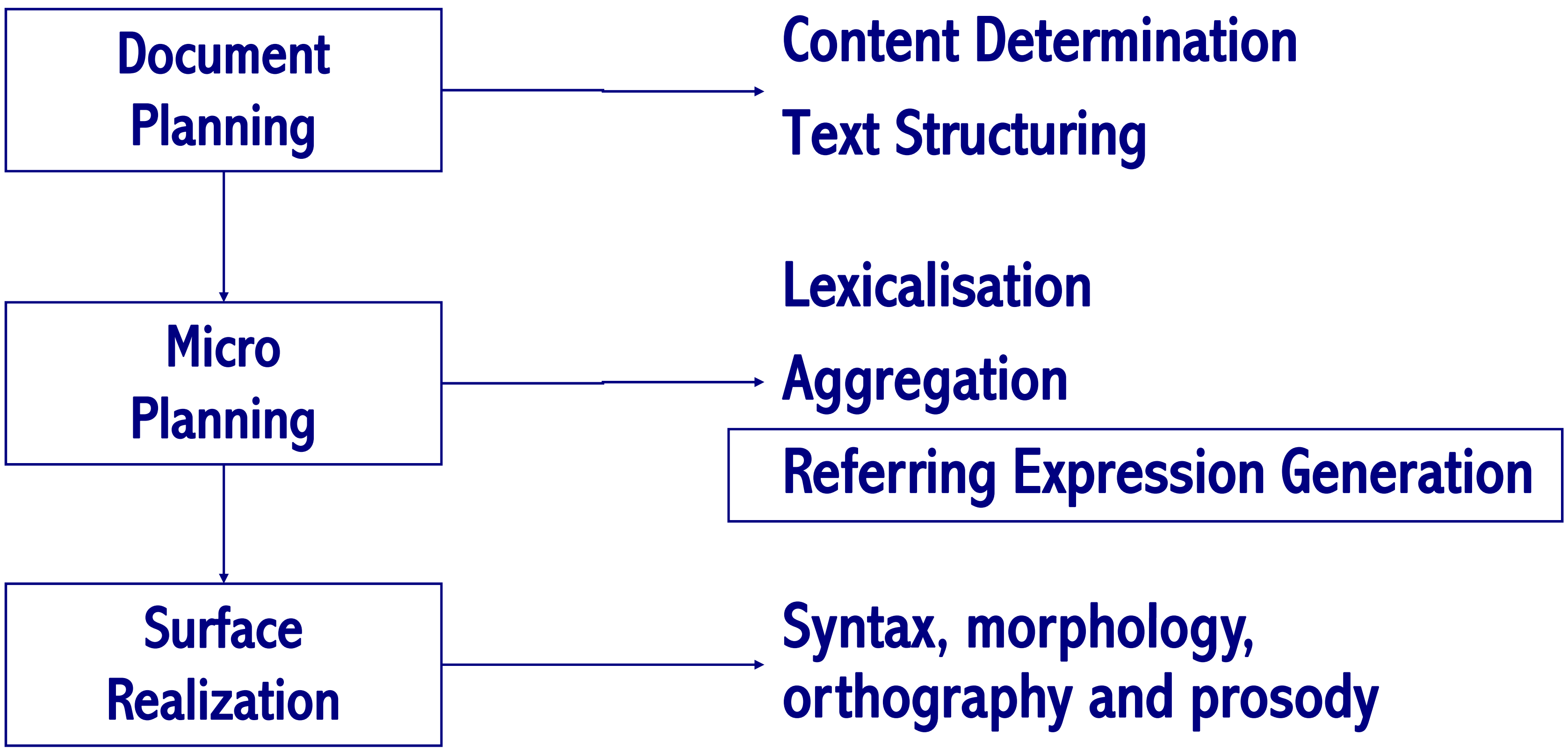
John kissed Mary.

Mary was kissed by John.

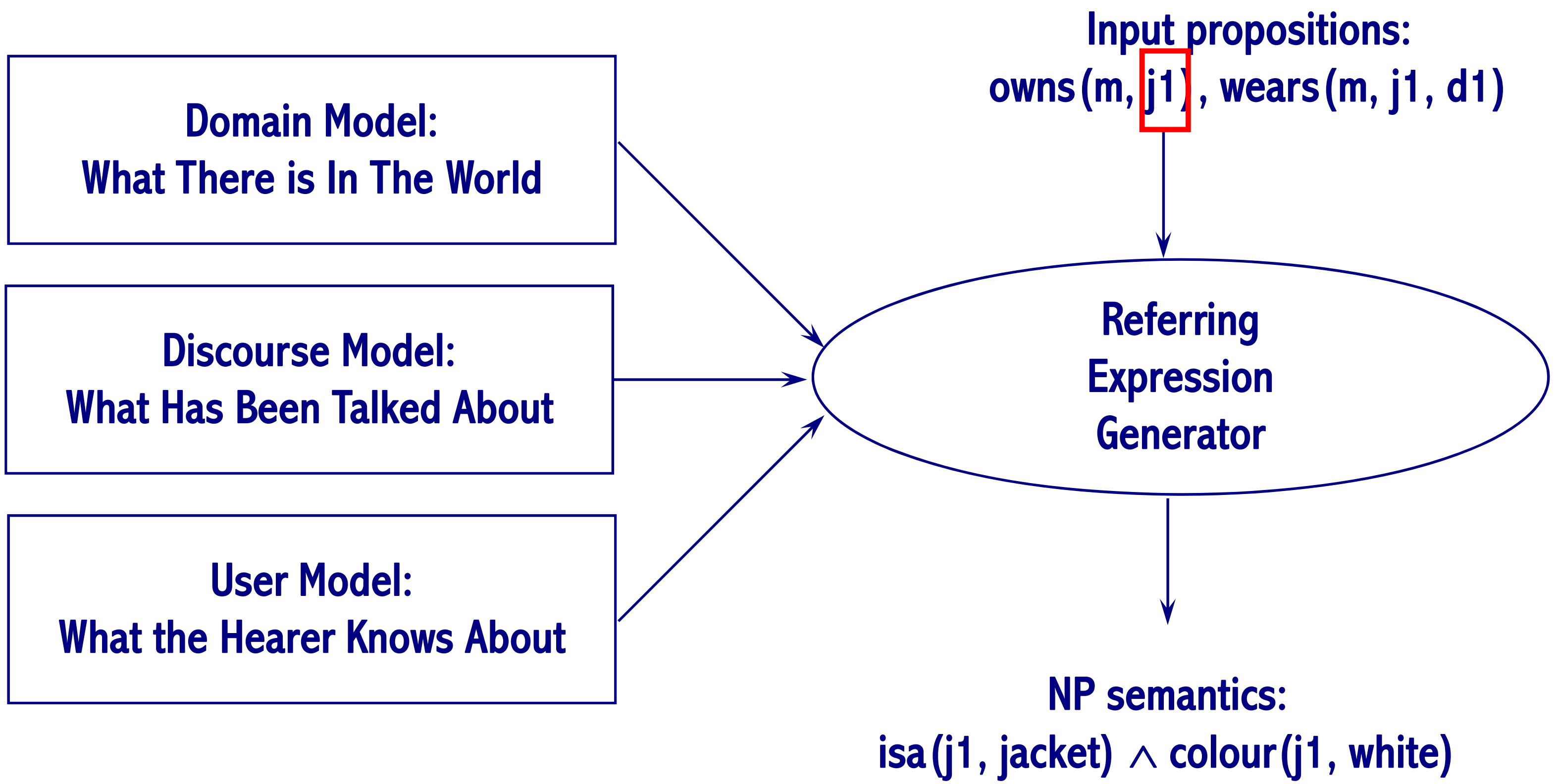
The NLGer's Position

- If we understand how and why texts are put together the way they are, we will be in a better position to take them apart
- Generation provides insights that should improve
  - Information extraction: working out what parts of a text are important
  - Text summarisation: working out how to replace incomplete references in extracted material
  - Machine translation: making choices that are appropriate to context

An Architecture for Generation



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 <u>it</u> on Sundays.
- Example 2:
  - owns(m, [j1+c1]) → Matt owns a white jacket and a white coat.
  - wears(m, j1, d) → He wears <u>the jacket</u> on Sundays.
- Example 3:
  - owns(m, [j1+j2]) →Matt owns a white jacket and a blue jacket.
  - wears(m, j1, d) → He wears <u>the white one</u> on Sundays.

Same

Different

Outline

- The Context: Natural Language Generation
- The Story So Far: Algorithm Development to Empiricism
- Challenges for the Future

The Consensus Problem Statement

The goal:

Generate a distinguishing description

Given:

- an intended referent;
- a knowledge base of entities characterised by properties expressed as attribute–value pairs; and
- a context consisting of other entities that are salient;

Then:

- choose a set of attribute–value pairs that uniquely identify the intended referent

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]

An Example

- Suppose x1 is the intended referent:

Entity	Type	Size	State
x1	dog	small	mangy
x2	dog	large	scurvy
x3	cat	small	mangy

An Example

- Choose 'mangy' to rule out x2:

Entity	Type	Size	State
x1	dog	small	$\boxed{\text{mangy}}$
x2	dog	large	scurvy
x3	cat	small	mangy

An Example

- Choose ‘mangy’ to rule out x2:

Entity	Type	Size	State
x1	dog	small	$\boxed{\text{mangy}}$
~~x2~~	~~dog~~	~~large~~	~~scurvy~~
x3	cat	small	mangy

An Example

- Choose ‘dog’ to rule out x3:

Entity	Type	Size	State
x1	$\boxed{\text{dog}}$	small	mangy
~~x2~~	~~dog~~	~~large~~	~~scurvy~~
x3	cat	small	mangy

An Example

- Choose ‘dog’ to rule out x3:

Entity	Type	Size	State
x1	$\boxed{\text{dog}}$	small	mangy
~~x2~~	~~dog~~	~~large~~	~~scurvy~~
~~x3~~	~~cat~~	~~small~~	~~mangy~~

- The result is ‘the mangy dog’
- ‘The small dog’ is also a distinguishing description.

Problem #1: Computational Complexity

- The algorithm does not guarantee to find a minimal distinguishing description [Reiter 1990]

Problem #2: No User Model

- The algorithm assumes that all properties are equal: it is only the relative discriminatory power, and nothing else, that causes a particular property to be selected.
- Some properties are more useful than other properties which have the same discriminatory power.

A talks to B on the tram:

A: Which stop do I want for the cinema?

B: You should take the stop before mine.

Problem #3:
It's Not What People Do

- Context Set = b1, c1, c2
- Intended Referent = b1
- Domain Model:
  - bird(b1), white(b1)
  - cup(c1), black(c1)
  - cup(c2), white(c2)
- Typical description: 'the white bird'

A Response: The Incremental Algorithm

Initial Conditions:

- $C_r = \langle$all entities$\rangle$; $P = \langle$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]

The Key Property of the Incremental Algorithm

- Principle distinction between:
  - the way choices are made (domain independent)
  - the choices available (domain dependent)

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 <u>lacks</u> some property?
  - 'the dog that isn't a poodle'
- Extensions:
  - Sets [Stone 2000]
  - Negation and Disjunction [van Deemter 2002]:

More Algorithm Development: A Selection

- Integration of linguistic reference and pointing [Reithinger 1987]
- Generating quantifiers [Creaney 1996]
- Integration of constraint-based and incremental approaches [Horacek 1996]
- Incorporation of linguistic constraints to ensure expressibility [Horacek 1997]
- Simultaneous semantic and syntactic construction [Stone and Webber 1998]
- Incorporation of a treatment of salience [Krahmer and Theune 2002]
- Extension to sets [Gatt 2007]

Consolidation and Dissent: Unifying Frameworks

- Reconceptualisation as subgraph construction [Krahmer et al 2001, 2002]
- Reconceptualisation as parameterised search [Bohnet and Dale 2005]

Current Preoccupations in The Field: Empiricism and Evaluation

- How do our algorithms compare with what people do?
- How do our algorithms compare against each other?
- Not covered here: Anja Belz's work on Shared Task Evaluation Campaigns (see http://www.itri.brighton.ac.uk/research/reg08/)

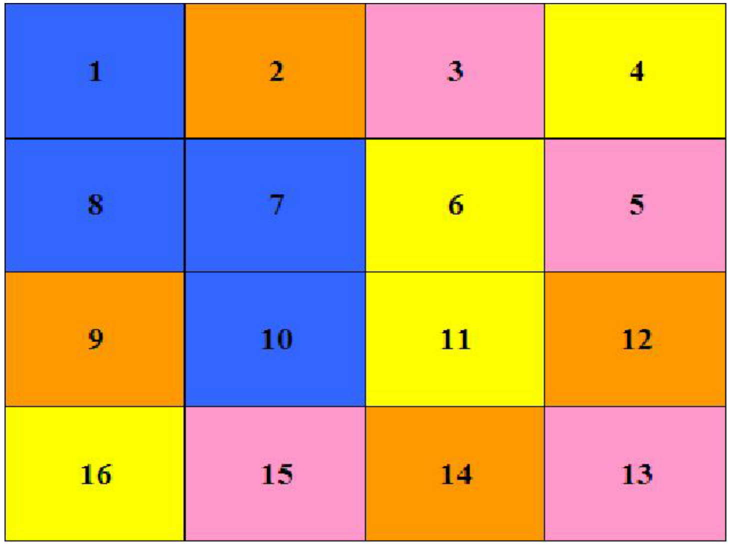
What Do People Do?

- The HCRC Map Task Corpus [Varges 2005]
- The Macquarie Drawers Corpus [Viethen and Dale 2006]
- The TUNA Corpus [van Deemter et al 2006]
- The Macquarie Blocks Corpus [Viethen and Dale 2008]

Experiment #1: The Macquarie Drawers Corpus

The Drawers Domain [Viethen + Dale 2006]:

- a grid of 4 × 4 filing cabinet drawers
- each has a number in the range 1–16
- four drawers each are blue, yellow, pink and orange

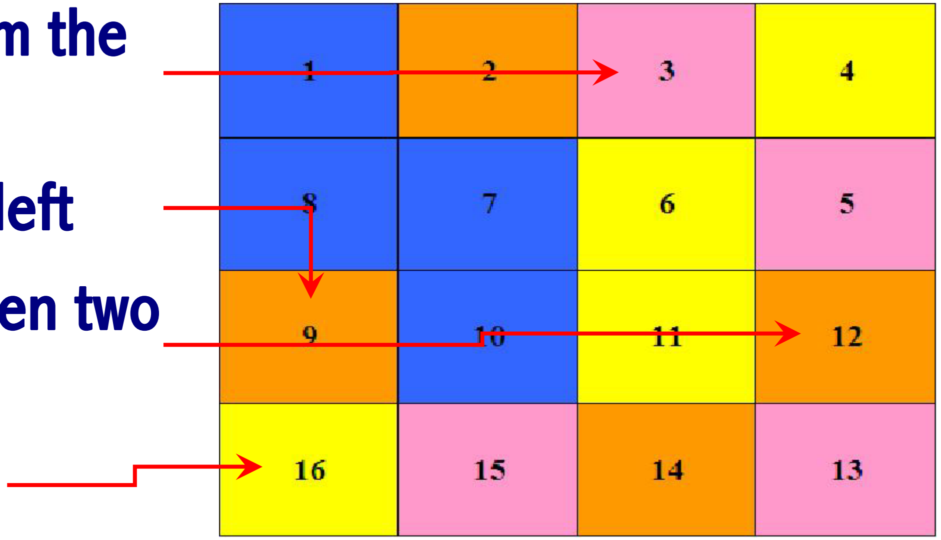


Task:

- Given the number of a drawer, describe it to an onlooker without mentioning any of the numbers
- 20 participants → 140 descriptions (between 3 and 12 per drawer)

Some Human-generated Descriptions

- D3: the top drawer second from the right
- D9: the orange drawer on the left
- D12: the orange drawer between two pink ones
- D16: the bottom left drawer

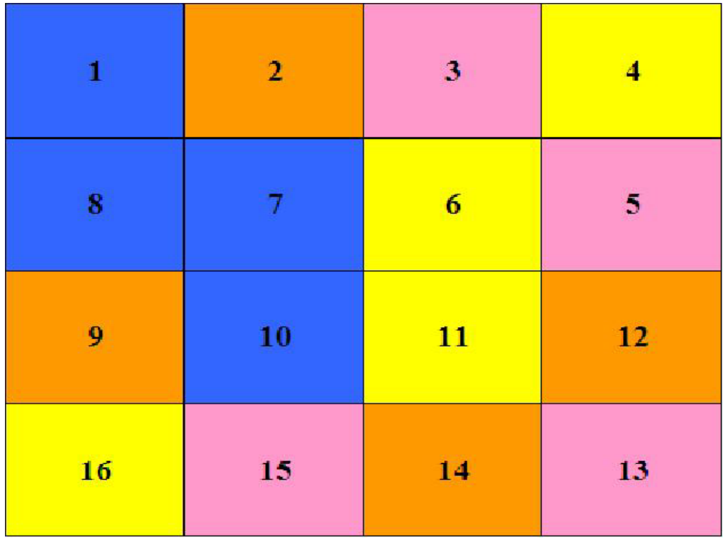


Characteristics of the Data Set

- People don't always produce minimal descriptions:
  - Minimal Descriptions: 75.4% (89)
  - Redundant Descriptions: 24.6% (29)
- People rarely use relational descriptions:
  - One-place Predicates Only: 87.3% (103)
  - Relational Descriptions: 12.7% (15)

Redundant Descriptions

- D6: the yellow drawer in the third column from the left second from the top
- D1: the blue drawer in the top left corner
- D14: the orange drawer below the two yellow drawers

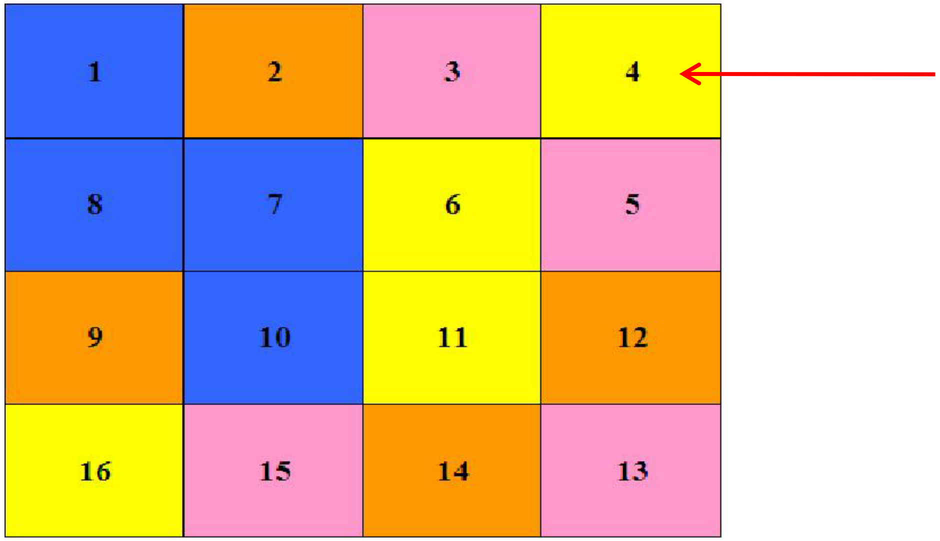


How Do Our Algorithms Fare?

Description type / Algorithm	Overall	Minimal	Redundant	Relational
Greedy [Dale 1989]	79.6%	100%	31.0%	-
Incremental [Dale + Reiter 1995]	95.1%	100%	82.8%	-
Relational [Dale + Haddock 1991]	0%	0%	0%	0%

The Problem with Relations

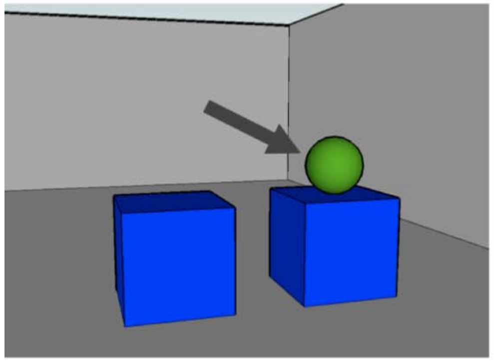
- The Dale and Haddock algorithm prefers relations over other potential elements to include:
  - the drawer above the drawer above the drawer above the pink drawer



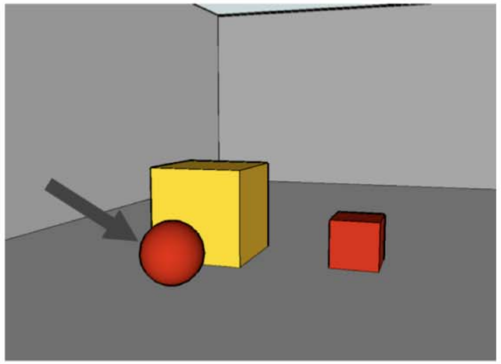
Experiment #2: The Macquarie Blocks Corpus

- 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

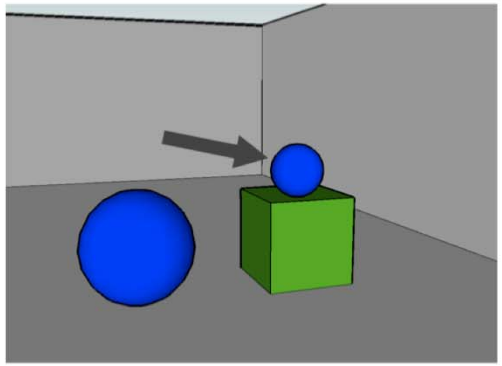
The Macquarie Blocks Corpus



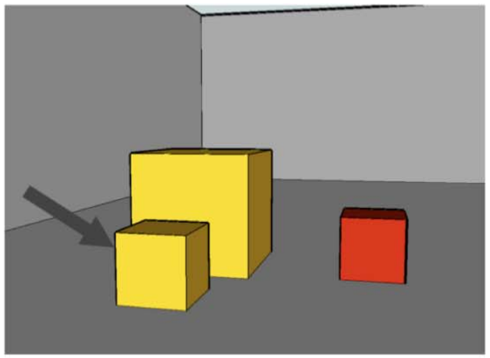
1



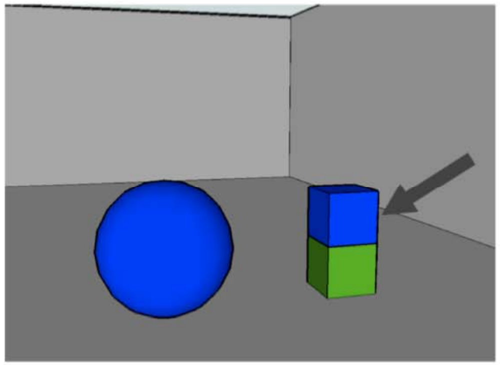
2



3



4



5

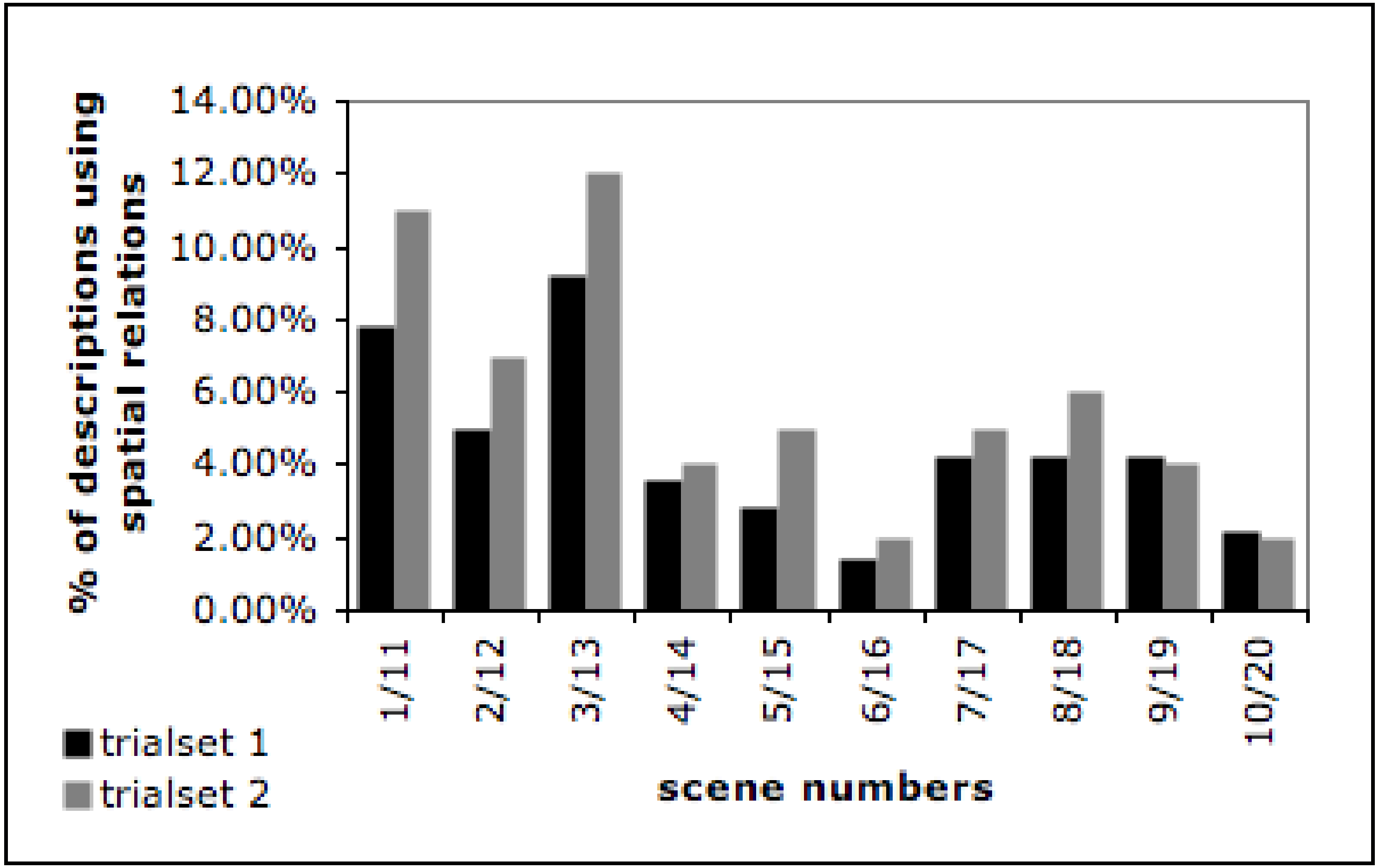
The Data

- 740 descriptions
- Data for 11 subjects removed:
  - 1on participant’s request
  - 1because subject was colour blind
  - 9 because of apparent misunderstanding of the task
- Final set = 630 descriptions

Some Results

- 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

Variation Across Duration of Trial



Interim Conclusions

- Spatial relations are used even when unnecessary
- There is a training effect: people become more confident in <u>not</u> using relations
- Landmark salience encourages use of relations

Consequences for Algorithm Development

- Need to incorporate scope for individual variation: perhaps a ‘risky’ versus ‘cautious’ parameter? [Carletta 1992]
- Need finer-grained account of characteristics of properties in the domain:
  - the ease with which a potential landmark can be distinguished, and its visual salience
  - the type of spatial relation between the target and a potential landmark
  - the ease with which the target can be described without the use of spatial relations

Outline

- The Context: Natural Language Generation
- The Story So Far: Algorithm Development to Empiricism
- Challenges for the Future

So Where Are We At Now?

- A number of base algorithms within the standard framework
- Some tentative explorations into other ways of thinking about the problem; extensions to accommodate sets, negation, disjunction, bridging reference, salience, pointing, linguistic constraints, quantifiers … lots of pieces that haven't yet been glued together
- An evolving understanding of the role of evaluation and empirical data gathering

Challenges for the Future

1. Consolidation
2. Use in Applications
3. Broadening the Story: Other Uses of Reference

Challenge #1: Consolidation

- We have many piecemeal algorithms for different aspects of referring expression generation
- Nobody so far has glued them altogether

A Skeletal Algorithm

Given an intended referent x:

begin

if x is in focus

then use a pronoun

elseif x has been mentioned already

then build a definite noun phrase

else build an initial indefinite reference

end

But What About Pronouns?

Given an intended referent x:

begin

if x is in focus

then use a pronoun

elseif x has been mentioned already

then build a definite noun phrase

else build an initial indefinite reference

end

And What About Initial Reference?

Given an intended referent x:

begin

if x is in focus

then use a pronoun

elseif x has been mentioned already

then build a definite noun phrase

else build an initial indefinite reference

end

Consolidation Challenges

- Covering the ‘Identification Space’
  - Pronominal Reference
  - Initial Reference
- Scaling up syntactic and semantic coverage
- Integration of experimental findings

Challenge #2: Use in Applications

- Referring Expression Generation is still a theory-bound enterprise
- But there is real scope for practical applications:
  - Entity description in tailored instructions
  - Landmarks and directions in route descriptions
  - Entity references in automatically-generated summaries

Instructions

1. Remove the modem card from its packaging.
2. Align the card to the matching ISA or PCI slot.
3. Remove the slot cover to allow the modem ports to be accessible from the outside of the computer.
4. Carefully insert the card into the slot and push firmly into place. Secure the card with a screw in the metal tab.
5. Replace the cover, plug in the power cord, and turn on the computer.

Route Descriptions

- A couple of kilometers after the M2 turn off is Herring Road, at the top of a hill.
- You'll pass through a built up suburb with lots of shops called St Ives; then you'll go under the Pacific Highway, at which point the road changes its name to Ryde Road.
- After going downhill and up again, you'll start going down hill into a valley through which the Lane Cover River runs; the road's called Lane Cove Road at this point.
- Turn left at the first set of lights, which will take you into the university.

Entity Reference in News Stories

Morgan Stockbroking Ltd said it was recommending newly-listed equipment hire group Coates Hire Ltd as a buy, reflecting good growth prospects. "The company is attractively priced based on 1997 fundamentals," analyst John Clifford said in a report. Coates listed this month after the sale of Australian National Industries Ltd's 100 percent holding had a balance sheet "comfortably geared" at 46 percent and interest cover forecast to rise to eight times in the year ended June 30, 1997 from 6.7 times in 1995/96, Clifford said.

Challenge #3: The Discourse Functions of Reference

- There is more to reference than attribute selection for discrimination
- The role that a noun phrase plays in a discourse impacts on the attribute selection process
  - Maintaining focus
  - Setting the stage for subsequent reference
  - Contrasting one entity with another
  - Highlighting specific properties

Adding Discourse Purpose to Referring Expression Generation

- We already have theory of discourse structure that has been well-explored in NLG: Rhetorical Structure Theory
- Consider each element of a nominal expression as being licenced by some rhetorical function or purpose
  - Distinguishing from potential distractors is just one function
- The challenge: to catalog the inventory of rhetorical functions that surface in nominal expressions
- Likely to be domain- and genre-specific

Conclusions

- Referring Expression Generation is the most well-defined and developed subfield of NLG … but we've only just got started
- There are real near-term practical applications that can benefit:
  - Instruction Manuals and Technical Support
  - Route Description
  - Entity Reference in Document Summarisation
- Natural language generation remains the best theoretical perspective for understanding how language really works