Generating Referring Expressions in Open Domains

Advaith Siddharthan & Ann Copestake

as372@cs.columbia.edu & aac10@cl.cam.ac.uk

Structure of Talk—1

- Motivation
- Attribute Selection
  - The Incremental Algorithm (IE) (Reiter and Dale, 1992)
  - Various Problems
  - Our Approach
  - A Comparison
- Relations
- Nominals
- Evaluation
- Conclusions

Motivation

<ins>A former ceremonial officer from Derby</ins>, *who was at the heart of Whitehall's patronage machinery*, says there is a general review of the state of the honours list every five years or so.

$$\downarrow$$

<ins>A former ceremonial officer from Derby</ins> says there is a general review of the state of the honours list every five years or so. <ins>This former officer</ins> was at the heart of Whitehall's patronage machinery.

The Incremental Algorithm (IA)

- Reiter and Dale (1992)
- Representation of Entities:

$$\begin{bmatrix} \texttt{type} & dog \\ \texttt{size} & small \\ \texttt{colour} & black \end{bmatrix} \quad \begin{bmatrix} \texttt{type} & dog \\ \texttt{size} & large \\ \texttt{colour} & black \end{bmatrix}$$

- Input:
  - intended referrent (AVM)
  - contrast set (AVMs)
  - `*preferred-attributes*` list
    eg: `[colour, size, shape,...]`

IA continued

$$e1 = \begin{bmatrix} \texttt{type} & dog \\ \texttt{size} & small \\ \texttt{colour} & black \end{bmatrix} \qquad e2 = \begin{bmatrix} \texttt{type} & dog \\ \texttt{size} & large \\ \texttt{colour} & black \end{bmatrix}$$

`*preferred-attributes*` = {colour, size, shape}

- Incremental Step:
  Add an attribute from `*preferred-attributes*` that rules out at least one entity in the contrast set.
- End Condition:
  All the entities in the contrast set have been ruled out.
  OR
  All the attributes have been used up

Justification

The psycholinguistic justification for the incremental algorithm:

- Humans build up referring expressions incrementally.
- Humans often use sub-optimal expressions.
- There is a preferred order in which humans select attributes eg. colour>shape>size...

Problems with the IA

Assumptions:

- A classification scheme for attributes exists
- The values that an attribute can take are mutually exclusive.

  eg: e1 = {big dark dog}   e2 = { huge black dog}

- Linguistic realisation of attributes are unambiguous

$$e1 = \begin{bmatrix} \texttt{type} & \textit{president} \\ \texttt{age} & \textit{old} \\ \texttt{tenure} & \textit{present} \end{bmatrix} \qquad e2 = \begin{bmatrix} \texttt{type} & \textit{president} \\ \texttt{age} & \textit{young} \\ \texttt{tenure} & \textit{past} \end{bmatrix}$$

Our Approach

- Measures the relatedness of adjectives
- Works at the level of words, not their semantic labels.
- Treats discriminating power as only one criteria for selecting attributes
- Allows for the easy incorporation of other considerations:
  - reference modification
  - reader's comprehension skills

Discriminating Power

How useful is an adjective for referencing an entity?

We define three quotients:

- Similarity Quotient (SQ)
- Contrastive Quotient (CQ)
- Discriminating Quotient (DQ)

Similarity Quotient (SQ)

- Quantifies how similar an adjective (a_o) is to adjectives describing distractors
- Transitive WordNet synonymy
- We form the Sets:
  - S_1: WordNet synonyms of a_o
  - S_2: WordNet synonyms of members of S_1
  - S_3: WordNet synonyms of members of S_2
- For each adjective (a_i) descibing each distractor:
  - if a_i is in S_1, $SQ+ = 4$
  - else, if a_i is in S_2, $SQ+ = 2$
  - else, if a_i is in S_3, $SQ+ = 1$

Contrastive Quotient (CQ)

- Quantifies how contrastive an adjective (a_o) is to adjectives describing distractors
- Transitive WordNet antonymy
- We form the Sets:
  - A_1: WordNet antonyms of a_o
  - A_2: WordNet synonyms of members of A_1
    + WordNet antonyms of members of S_1
  - A_3: WordNet synonyms of members of A_2
    + WordNet antonyms of members of S_2
- For each adjective (a_i) descibing each distractor:
  - if a_i is in A_1, $CQ+=4$
  - else, if a_i is in A_2, $CQ+=2$
  - else, if a_i is in A_3, $CQ+=1$

Discriminating Quotient (DQ)

- An attribute with high SQ has bad discriminating power.
- An attribute with high CQ has good discriminating power.
- We define the Discriminating Quotient (DQ) as

$$DQ = CQ - SQ$$

- We now have an order (decreasing DQs) in which to incorporate attributes

Example—1

$$e1(\textit{referent}) = \begin{bmatrix} \texttt{type} & \textit{president} \\ \texttt{age} & \textit{old} \\ \texttt{tenure} & \textit{current} \end{bmatrix} \qquad e2(\textit{distractor}) = \begin{bmatrix} \texttt{type} & \textit{president} \\ \texttt{age} & \textit{young} \\ \texttt{tenure} & \textit{past} \end{bmatrix}$$

- Assume we want to refer to `e1`.
- Following a typing system, comparing the `age` attribute would rule out `e2`
- We would end up with the old president that is ambiguous.

attribute	distractor	CQ	SQ	DQ
old	`e2`{young, past}	4	4	0
current	`e2`{young, past}	2	0	2

Example—2

We have four dogs in context: `e1`(a large brown dog), `e2`(a small black dog), `e3`(a tiny white dog) and `e4`(a big dark dog).

To refer to `e4`:

attribute	distractor	CQ	SQ	DQ
big	`e1`{large, brown}	0	4	-4
big	`e2`{small, black}	4	0	4
big	`e3`{tiny, white}	1	0	1
				1
dark	`e1`{large, brown}	0	0	0
dark	`e2`{small, black}	1	4	-3
dark	`e3`{tiny, white}	2	1	1
				-2

the big dark dog

Example—3

We have four dogs in context: `e1`(a large brown dog), `e2`(a small black dog), `e3`(a tiny white dog) and `e4`(a big dark dog).

To refer to `e3`:

attribute	distractor	CQ	SQ	DQ
tiny	`e1`{large, brown}	1	0	1
tiny	`e2`{small, black}	0	1	-1
tiny	`e4`{big, dark}	1	0	1
				1
white	`e1`{large, brown}	0	0	0
white	`e2`{small, black}	4	0	4
white	`e4`{big, dark}	2	0	2
				6

the white dog

Justification -Psycholinguistic

The psycholinguistic justification for the incremental algorithm:

1. Humans build up referring expressions incrementally.
2. There is a preferred order in which humans select attributes eg. colour$>$shape$>$size...

Our algorithm:

- Is also incremental but differs from premise 2
- Assumes that speakers pick out attributes that are distinctive in context
- Averaged over contexts, some attributes have more discriminating power than others (largely because of the way we visualise entities)
- Premise 2 is an approximation to our approach.

Justification -Computational

N = Max number of entities in the contrast set

n = Max number of attributes per entity

Incremental Algo	Our Algorithm	Optimal Algo[1]
$O(nN)$	$O(n^2N)$	$O(n2^N)$

[1] such as Reiter (1990)

Other Considerations

- Discriminating power is only one of many reasons for selecting an attribute.

Reference Modification

- Attributes can be reference modifying:
  - `e1` = *an alleged murderer*
  - *alleged* modifies the reference *murderer*
  - *alleged* does not modify the referent `e1`
- We handle reference modifying adjectives trivially by adding a positive weight to their DQs.
- This has the effect of forcing that attribute to be selected in the referring expression.

Reading Skills

- Uncommon adjectives have more discriminating power than common adjectives.
- However, they are more likely to be incomprehensible to people with low reading ages.
- Giving uncommon adjectives higher weights will generate referring expressions with fewer, though harder to understand, adjectives.
- Giving common adjectives higher weights will generate referring expressions with many simple adjectives.

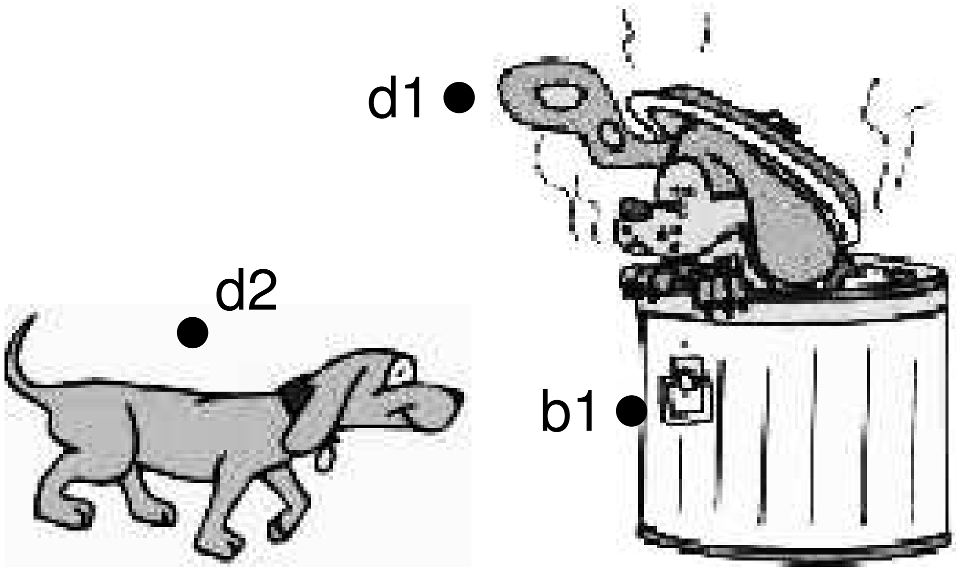
Contrast Sets and Salience

- The incremental algorithm assumes the availability of a contrast set of distractors
- The contrast set, in general, needs to take context into account
- Krahmer and Theune (2002) propose an extension to the incremental algorithm which treats the contrast set as a combination of a discourse domain and a salience function.
- Incorporating salience into our algorithm is trivial
  - We computed SQ and CQ for an attribute by adding $w \in \{4, 2, 1\}$ to them each time a distractor's attribute was discovered in a synonym or antonym list.
  - We can incorporate salience by weighting w with the salience of the distractor whose attribute we are considering.
  - This will result in attributes with high discriminating power with regard to more salient distractors getting selected first in the incremental process.

To Summarise...

- Reference generation belongs in the realisation module, not in microplanning.
- Adjective classification is *unnatural* and infeasable
- Context matters
- Attribute selection is possible regardless
- Discriminating power is only one of many criteria

Relations



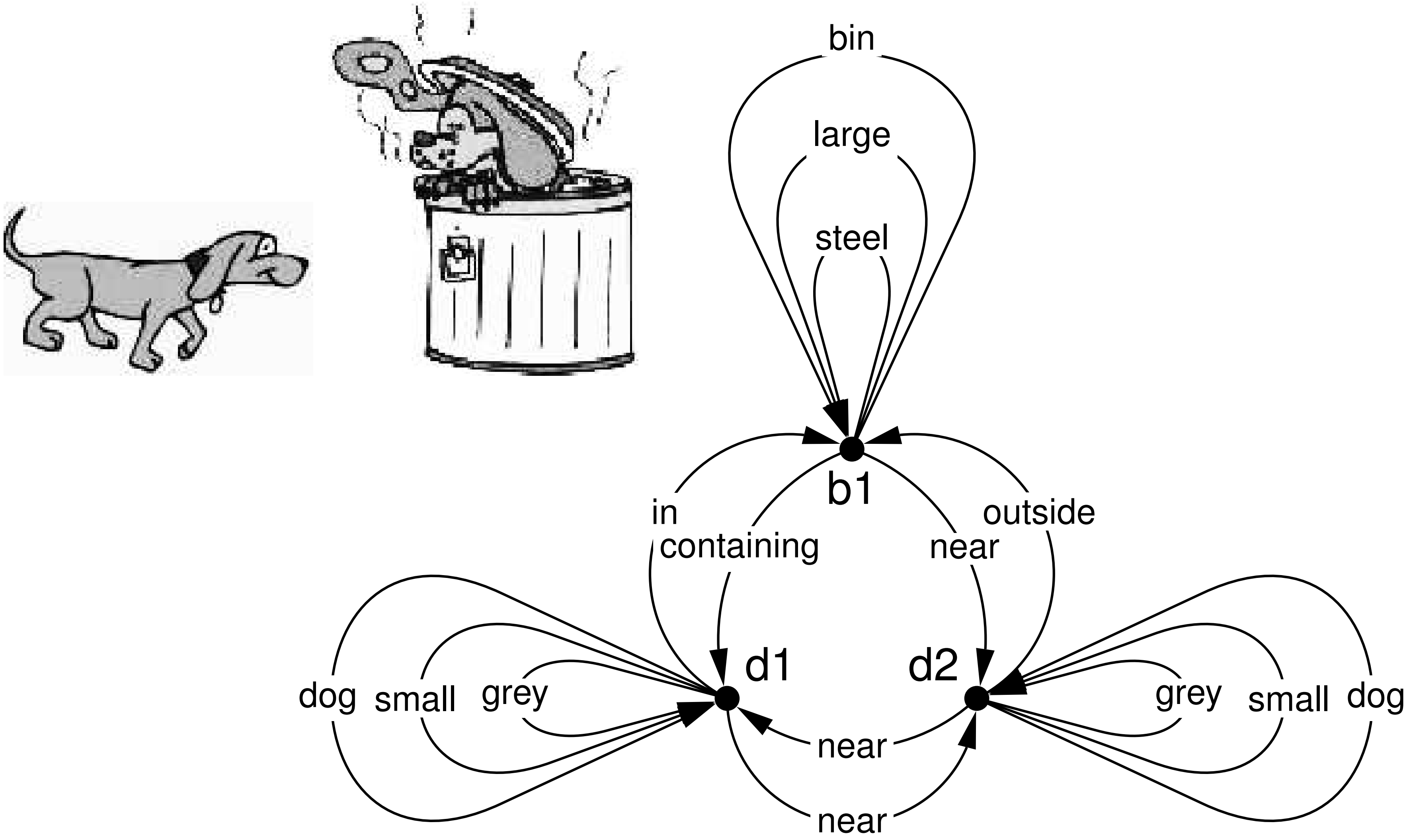
$$d1 = \begin{bmatrix} \texttt{head} & dog \\ \texttt{attrib} & [small, \\ & grey] \\ \texttt{in} & b1 \\ \texttt{near} & d2 \end{bmatrix} \quad d2 = \begin{bmatrix} \texttt{head} & dog \\ \texttt{attrib} & [small, \\ & grey] \\ \texttt{outside} & b1 \\ \texttt{near} & d1 \end{bmatrix} \quad b1 = \begin{bmatrix} \texttt{head} & bin \\ \texttt{attrib} & [large, \\ & steel] \\ \texttt{containing} & d1 \\ \texttt{near} & d2 \end{bmatrix}$$

Relations

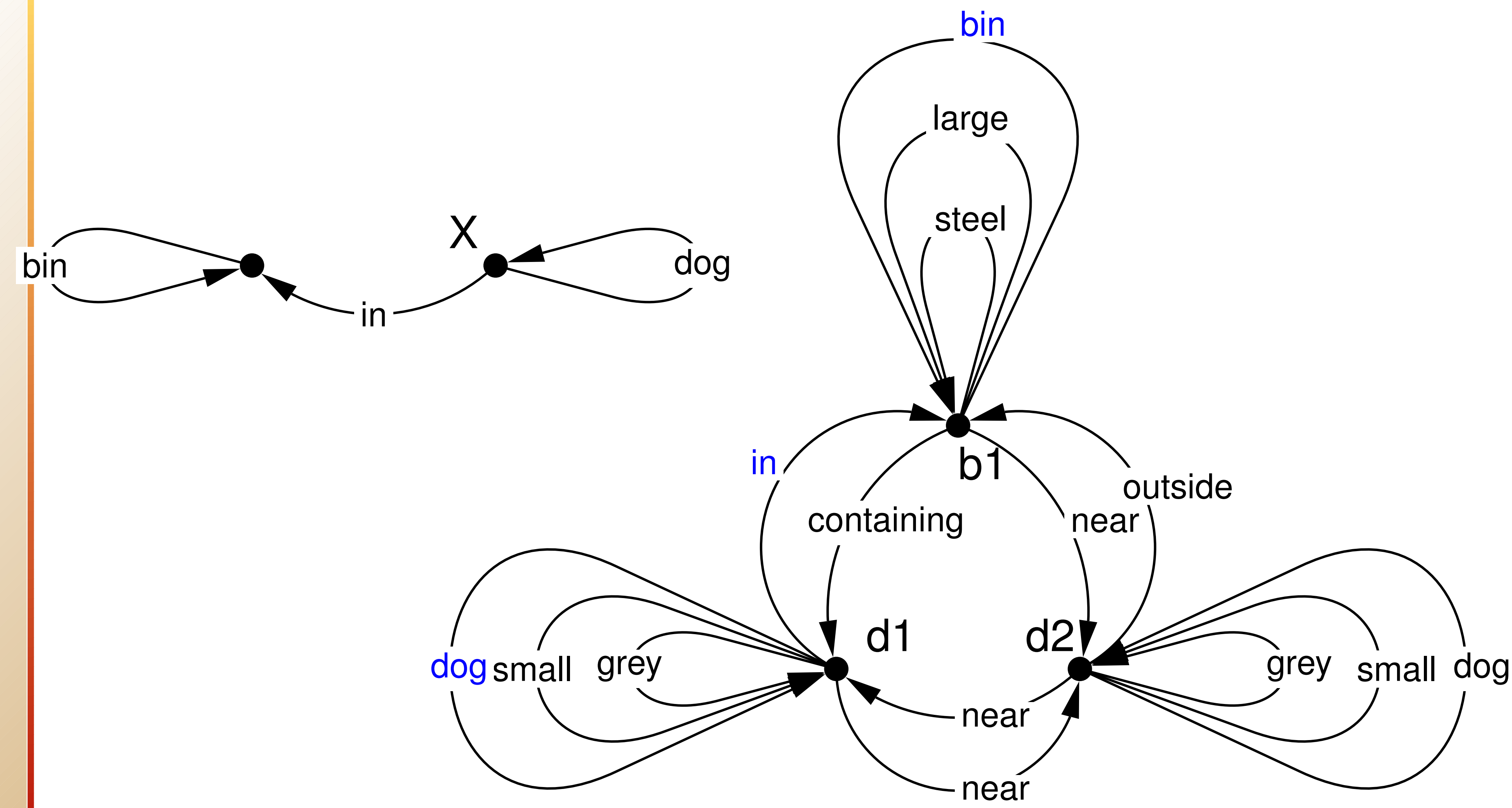
- *attributes* describe an entity *(the small grey dog)*
- *relations* relate an entity to other entities *(the dog in the big bin)*
- The IA does not consider relations and the referring expression is constructed out of only attributes.
- It is difficult to imagine how relational descriptions can be incorporated in the incremental framework of the IA
- Dale and Haddock (1991) allows for relational descriptions but involves exponential global search.
- Our approach computes the order in which attributes are incorporated on the fly, by quantifying their utility through DQ.
- We can compute DQ for relations in much the same way as we did for attributes

Graph Approach

Krahmer et al. (2003)



Graph Approach



Calculating DQ for Relations

To compute the three quotients for the relation $[prep_o\ e_o]$:

- We consider each entity e_i in the contrast set in turn.
- If e_i does not have a $prep_o$ relation $CQ+ = 4$
- If e_i has a $prep_o$ relation:
  - If the object of e_i's $prep_o$ relation is e_o then $SQ+ = 4$.
  - Else $CQ+ = 4$.
- For attributes, we defined $DQ = CQ - SQ$.
- For relations, we can define $DQ = (CQ - SQ)/length$
- Approximate $length$ as $length = 3 + n$ where n is number of distractors containing a $prep_o$ relation with a non-e_o object

Discourse Plans

- Attributes are usually used to *identify* an entity
- Relations, in most cases, serve to *locate* an entity
- Generating instructions for using a machine:

  switch on the red button on the top-left corner
- Generating directions for finding things

  The salt behind the corn flakes on the shelf above the fridge
- If the discourse plan requires preferential selection of relations or attributes, we can add a positive amount α to their DQs
- $DQ = (CQ - SQ)/length + \alpha$
- $length = 1$ for attributes
- By default, $\alpha = 0$ for both relations and attributes.

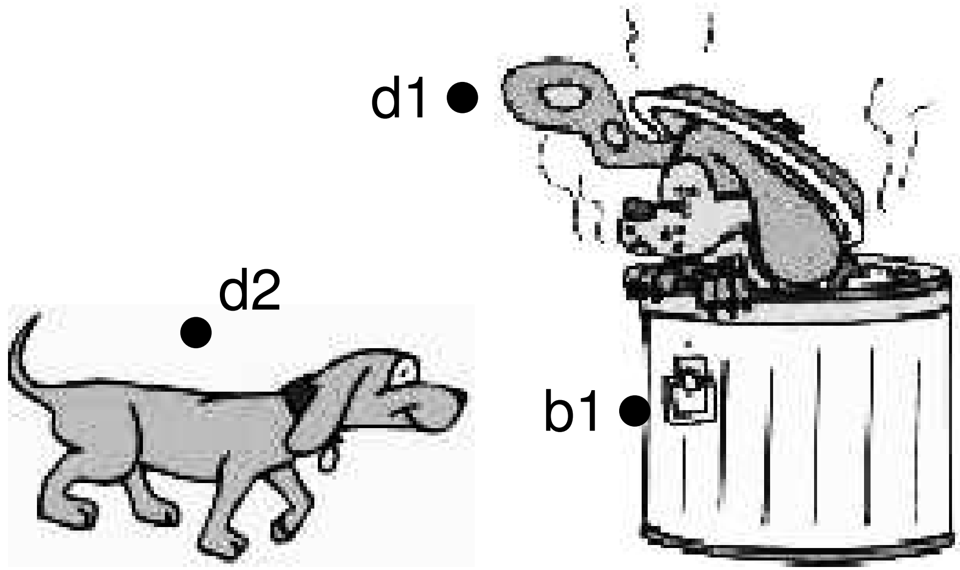
The Algorithm

To generate a referring expression for an entity:

- calculate DQs for all its attributes and approximate the DQs for all its relations.
- form the **preferred** list
- add elements of **preferred** till the contrast set is empty
  - straightforward for attributes
  - For relations, recursively generate the prepositional phrase first
    - check that it hasn't entered a loop

      the dog in the bin containing the dog in the bin...
    - generate a new contrast set for the object(bin)
    - recursively generate a referring expression for the object of the relation

An Example



$$d1 = \begin{bmatrix} \texttt{head} & dog \\ \texttt{attrib} & [small, \\ & grey] \\ \texttt{in} & b1 \\ \texttt{near} & d2 \end{bmatrix} \quad d2 = \begin{bmatrix} \texttt{head} & dog \\ \texttt{attrib} & [small, \\ & grey] \\ \texttt{outside} & b1 \\ \texttt{near} & d1 \end{bmatrix} \quad b1 = \begin{bmatrix} \texttt{head} & bin \\ \texttt{attrib} & [large, \\ & steel] \\ \texttt{containing} & d1 \\ \texttt{near} & d2 \end{bmatrix}$$

An Example

Referring Expression for `d1`

- *ContrastSet* = [d2]
- $DQ_{small} = -4$, $DQ_{grey} = -4$
  $$DQ_{[in\ b1]} = 4/3,\ DQ_{[near\ d2]} = 4/4$$
- **preferred** = [[in `b1`], [near `d2`], small, grey]
- iteration 1: [in `b1`]
  - *ContrastSet* is empty
  - return {bin}
- add the PP [in the {bin}] to RE
- *ContrastSet* is now empty
- return {[in the {bin}], dog}

Nominals

- Nominals introduced through relations can also be introduced attributively

    1. professor at Columbia $\leftrightarrow$ Columbia professor
    2. novel by Archer $\leftrightarrow$ Archer novel
    3. president of IBM $\leftrightarrow$ IBM president
    4. company from East London $\leftrightarrow$ East London company
    5. church in Paris $\leftrightarrow$ Paris church

- We need to compare nominal attributes with the objects of relations.
- We also need to extend the algorithm for calculating DQ for a relation

An Example

Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents, *which precedes the full purchasing agents report* ***that is due out today*** *and gives an indication of what the full report might hold.*

$$e_o = \begin{bmatrix} \texttt{head} & \textit{report} \\ \texttt{by} & \begin{bmatrix} \texttt{head} & \textit{agents} \\ \texttt{attrib} & [\textit{Chicago}, \\ & \textit{purchasing}] \end{bmatrix} \end{bmatrix}$$

$$e_1 = \begin{bmatrix} \texttt{head} & \textit{report} \\ \texttt{attributes} & [\textit{full, purchasing, agents}] \end{bmatrix}$$

Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. The Chicago report precedes the full purchasing agents report and gives an indication of what the full report might hold. The full report is due out today.

Evaluation

- Notoriously difficult!
- Existing algos are domain specific
- Can't be compared easily
- No standard test sets
- In fact, no quality evaluations at all!

Evaluation

- Our Algo is open domain
- Evaluation possible on the Penn WSJ Treebank
  - We identified instances of referring expressions,
  - Then identified the antecedent & all the distractors in a four sentence window,
  - Then generated a referring expression for the antecedent, giving it a contrast-set containing the distractors
  - Compared with the ref exp. in the text.

Evaluation

- There were 146 instances of Ref Exps (noun phrases with a definite determiner) for which:
  - An antecedent was found for the referring expression.
  - There was at least one distractor in the discourse window.
  - The ref exp. had at least one attribute or relation.
- 81.5% Perfect!
- Many others seemed ok, some are hard to tell!
- eg: ref exp in WSJ = *the one-day limit*
  antecedent found = *the maximum one-day limit for the S&P 500 stock-index futures contract*
  Contrast set= *{the five-point opening limit for the contract, the 12-point limit, the 30-point limit, the intermediate limit of 20 points}*

  Our program generated = *the maximum limit*

Evaluation

- Examples of Wrong REs:

Noun Phrase	Generate Ref. Exp.
personal care products	care products
open end mutual funds	end funds
privately funded research	funded research

Conclusions

- Open Domain
- Selects attributes and relations that are distinctive in context
- Does not require adjective classification
- Incremental incorporations of relations
- Treatment of nominals
- Corpus-Based Evaluation!

References

Robert Dale and Nicholas Haddock. 1991. Generating referring expressions involving relations. In *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics (EACL'91)*, pages 161–166, Berlin, Germany.

Emiel Krahmer and Mariët Theune. 2002. Efficient context-sensitive generation of referring expressions. In Kees van Deemter and Rodger Kibble, editors, *Information Sharing: Givenness and Newness in Language Processing*, pages 223–264. CSLI Publications, Stanford,California.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.

Ehud Reiter and Robert Dale. 1992. A fast algorithm for the generation of referring expressions. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, pages 232–238, Nantes, France.

Ehud Reiter. 1990. The computational complexity of avoiding conversational implicatures. In *Proceedings of the 28th Annual Meeting of Association for Computational Linguistics (ACL'90)*, pages 97–104, Pittsburgh, Pennsylvania.

The Need for 3 Quotients

- Questions
  - Why do we need three different quotients?
  - In particular, what role does the synonymy quotient SQ play?
  - Why can't we perform the above analysis using only the contrastive quotient CQ?
- Answers
  - Our definition (CQ) of *contrastive* is too strict.
  - Combining SQ with AQ increases the robustness of the approach.
  - Computing antonyms transitively can give spurious results
  - But sensible results are found first