Search Methods in Natural Language Processing

(http://www.dfki.de/~horacek/search-NLP.html)

Helmut Horacek

Saarland University / DFKI

Language Technology, old bldg. 1.25
Tel: 85775-2450
email:helmut.horacek@dfki.de

No lectures on 9. & 16. May, 13. & 27. June

Possible day times for about 3 extra lectures

WHAT IS SEARCHING?

CHARACTERIZATION OF SEARCHING (in abstract terms)

Goal states

Set of desirable situations (typically defined by descriptive conditions)

Initial states

Set of existing situations (accessible from the start)

Solution

Path from an initial state to a goal state (by sequentially applying operators)

WHY SEARCHING IS NECESSARY

Problem size

Number of applicable operators (breadth)

Length of a path to the solution (depth)

Effort to uncover a solution

Cost of performing the search until successful

Problem structure

Considerable differences in dependency of approaches taken

WHEN IS SEARCHING BENEFICIAL ?

CONTRIBUTORS TO THE SUCCESS OF SEARCHING

Knowledge

Information about problem structure (concrete problem or domain knowledge)

Regularities present and/or heuristic assessements meaningful

Expectation

Solution properties known or can be evaluated

Technique

Systematic approach exploiting the above factors in the general case

WHAT SEARCHING IS NOT

Unsystematic approach

“I have a solution, but it does not fit to the problem”

Garfield

CHARACTERIZATION OF SEARCH METHODS
(in general terms)

Search strategy

Ways to explore the search space (essential differences among strategies)

Solution quality

If differences among goal state assessments exist, improvements are possible

Search effort

Proportional to the problem complexity; dependent on solution quality

NATURAL LANGUAGE PROCESSING AS SEARCH

Problem definition

Expressing the problem in terms of states and operators

Goal specification

Relating solution quality to search effort required

Search strategy

Adopting procedures that envision the goal within given specifications

(exploiting properties of natural languages, dependencies, …)

AN EXAMPLE TASK: GENERATING REFERRING EXPRESSIONS

Given

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

Goal specification

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

Search strategy

Incrementally build referring expressions and test 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

ALTERNATIVE OPTIONS

Problem space definition

Solution in terms of surface expressions or elements of the knowledge base

Goal specification

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

Search strategy

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

A GENERIC VIEW (Bohnet & Dale, IJCAI 2005)

Initial state

<empty expression, all distractors, all properties of the intended referent(s)>

Goal state

<chosen properties, no distractors, remaining properties>

Search strategy

combination of expansion, queuing, and cost computation

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)	**NP-hard**	$\approx n_a^{n_l}$
Greedy Heuristic (Dale 1989)	**polynomial**	$\approx n_a n_d n_l$
Local Brevity (Reiter 1990)	**polynomial**	$\approx n_a n_d n_l$
Incremental Algorithm (Dale,Reiter 1991)	**polynomial**	$\approx n_d n_l$

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

Influence of knowledge representation

Discriminatory power of some descriptors “delayed”

Search strategy

Limit embeddings – depth-first combined with breadth-first

Recursion of algorithm to related objects needs modification

Task embedding of descriptor selection

Realization potential on the surface must be anticipated

EXTENSION 2 – SETS OF OBJECTS (van Deemter 2000)

Functionality

Descriptors are extended to boolean combinations

Iteration over number of elements in a boolean combination

Search strategy

Breadth-first within iterative deepening

Assessment

Computationally efficient, but solution quality may be very low

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			•	•			•	•		•		•	

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

PARTITIONING INTENDED REFERENTS INTO SUBSETS

Transforming descriptions to reduce disjunctions

In $\wedge_{i=1,n}(\vee_{j=1,m} P_{ij})$ for several *i* non-atomic expressions likely

Picking one disjunction ($\vee_{j=1,mk} P_{kj}$) and transforming it according to distributivity

Yielding $\vee_{j=1,mk} (P_{kj} \wedge_{i=1,n\neq k}(\vee_{j=1,m} P_{ij}))$

Example

“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

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)

FURTHER ISSUES IN GENERATING REFERRING EXPRESSIONS

Descriptions with relations between objects

Expressions referring to sets of objects (including disjunctions of descriptors)

Multimodal referring expressions

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)

TWO INTERPRETATIONS OF SEARCHING

1. Performing systematic searches efficiently

Homogenous search spaces

(e.g., syntactic processing, statistical optimization)

2. Organizing a context-dependent construction process

Heterogenous search spaces

(e.g., natural language generation from a communicative intention)

PLAN FOR THE LECTURE

Introduction

Syntactic/surface-oriented methods
- ***Syntactic parsing***
- ***Syntactic generation***
- ***Discourse interpretation***

Machine translation methods
- ***Symbolic processing***
- ***Statistical processing***
- ***Stochastic generation***

Natural language generation (sentence planning)
- ***Aggregation***
- ***Generating referring expressions***

Architectural concerns
- ***The overall generation process - text planning***
- ***Specific issues in dialog systems***