On Intensional Answers to Database Queries

Helmut Horacek

German Research Center for Artificial Intelligence
Language Technology Division
Saarbrücken, Germany

Email: helmut.horacek@dfki.de

SEMAPRO 2021 -
The 15. International Conference on Advances in Semantic Processing

Barcelona, October 3, 2021 - October 7, 2021

Position

Researcher at DFKI Saarbrücken, Language Technology

Lecturer at University of the Saarland

Research interests

Natural language processing, specifically generation

Dialog systems

Proof presentation

Computational models of natural argument

Intelligent tutoring systems

Game playing

Projects I have contributed (back > 10 years)

FABEON - NL proof presentation (DFG project)

DIALOG - part of the Collaborative Research Centre on *Resource-Adaptive Cognitive Processes* (SFB 378) at University of the Saarland

Recent Projects at Language Technology DFKI

QT21: Quality Translation 21

EU Council Presidency Translator

Deep Learning for End-to-End-Applications in Language Technologie

feature learning without a priori knowledge about the language to be translated

Answers to Database Queries

Extensional answers (normal)

an enumeration of the data satisfying the query

Intensional answers (specific)

a discriminatory description of this data

Examples for intensional answers

„Which states have a capital?“

„All German states.“ (Integrity check, limited DB modeling)

„Which state does Missouri river pass through”?

„All the states that border South Dakota except Wyoming.“

including exception handling (Montana, South Dakota, North Dakota, Nebraska, Iowa, Kansas, Missouri)

Motivation

Expected benefits

Avoids potentially long enumerations

Highlights commonalities among items

May provide new (hopefully) useful information

Difficulties/ problem areas

Limited scope - potential extensions recently explored

Computational complexity

Situational suitability of descriptions may be problematic

Content

Research context

Cooperative answers

Deductive intentional answers (classical)

Inductive intentional answers (experimental)

Inductive intentional answers

Technical approaches

Heuristics, complexity, limitations

An equivalent linguistic task

The task - generating referring expressions

Approaches, algorithmic issues

The future

Algorithmic transfer

Issues of conceptual suitability

Cooperative Answers

Adequate treatment of empty responses

Detection of presupposition failures in queries

Stating the intermediate empty result

Answering slightly modified queries

Using knowledge bases

Approaching desired results by handling

near misses, slight variations (numerical, conceptual)

Overanswering

Anticipating follow-up questions

Adding conceptually related data

Intensional answers (classical)

Definitions

Circumscription of the semantics of the query

Sufficient conditions for the truth of the query

independent of content of the database

Issues

Exploiting subsumption, aiming at simplifications

Using vocabulary of interest to users

Based on reduced integrity constraints, deductive rules

Categories of Intensional Answers

Properties

Only intensional descriptions versus mixed forms

Independency/dependency of database content

Completeness in describing the extensional data

Approaches discussed here

Purely based on extensional data

Combinations (exceptions, near misses)

Extensions motivated by linguistic tasks

Intensional Answers built from Extensions

General approach

Based on inductive logic programming

Learning a concept that
subsumes only the extensions in the answer

Descriptions of extensions are built and
incrementally generalized

Choices in descriptions are made to reduce the complexity

Properties

Mixed - may include extensional data

Partial - not all/complete intensional descriptions generated

The Basic Idea (Hartfiel, Cimiano)

Ingredients of the approach

A knowledge base, with categories and properties of items

Procedural steps

- **Building a least general generalization, if possible**
- **Eliminating redundant clauses**
- **Checking coverage of only positive items**

Possible results

An intensional description

Failure to produce a (simple) description

Generalization Algorithm

$Ans = ExtensionalAanswers.getNext();$

$C = constructClause(Ans, KB);$

if $|ExtensionalAnswers| < 2$ **then**

  return C;

  while not all answers have been processed **do**

    $Ans' = ExtensionalAnswers.getNext();$

    $C' = constructClause(Ans', KB);$

    $c = LGG(C, C');$

    $Answers' = evaluateQuery(query(c), KB);$

    if $ExtensionalAnswers \cap Answers' \subseteq Answers'$ **then**

      return ϕ; $\triangleright$ no consistent clause can be foun

    $C'' = reduceClause(c, KB, ExtensionalAnswers);$

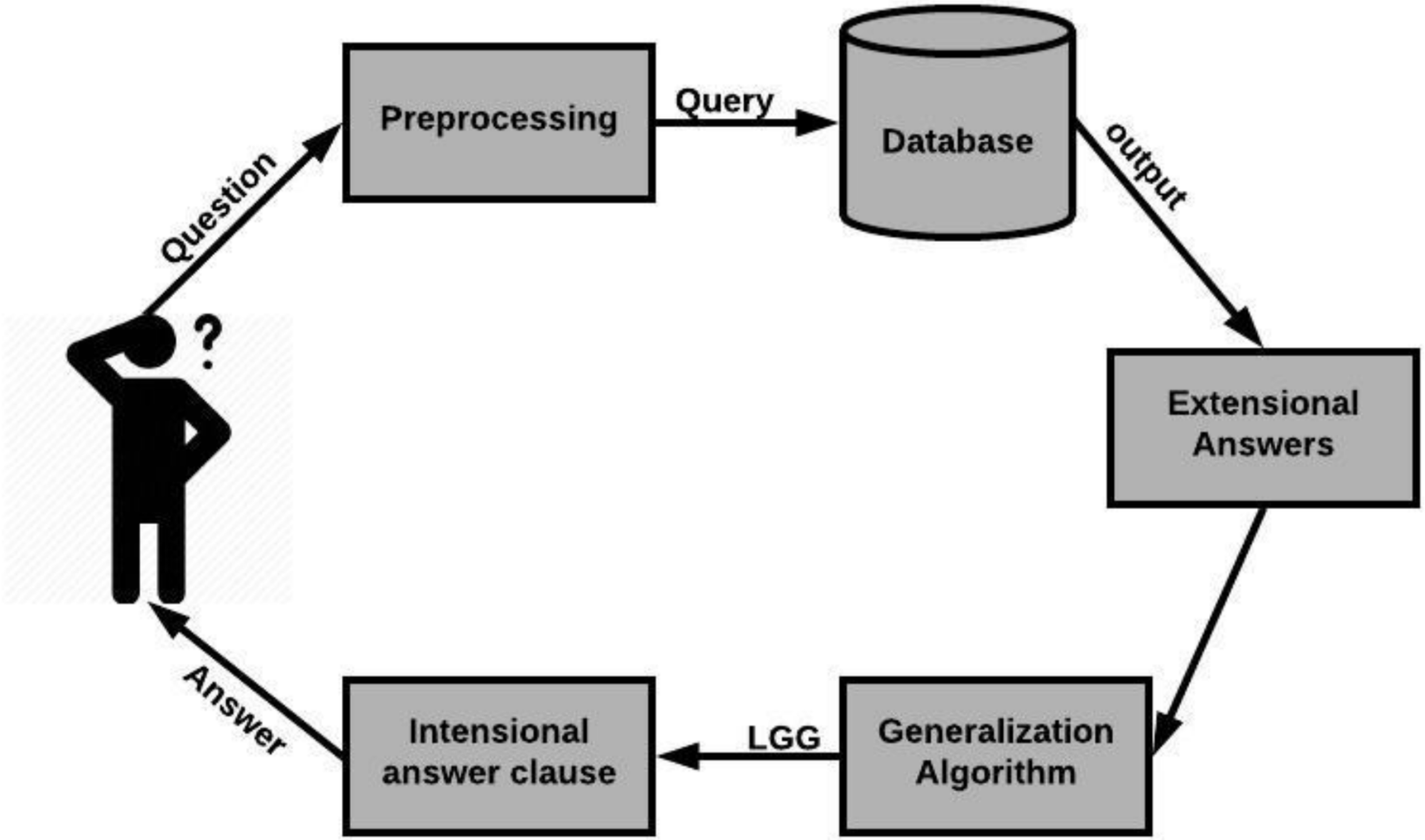
    $Answers'' = evaluateQuery(query(c''), KB);$

    if Answers”= Answers **then** $\triangleright$ i.e. C'' covers all answer

      return C'';

return $reduceClause(c, KB, Answers);$

General Architecture



Reduction Algorithm

ReduceClause(Clause c, KnowledgeBase KB, Set Answers)

$List\ literals = orderedLiterals(c)$; ▷ (in increasing order)
for i=1 to Literals **do**
    $c' = \text{remove}(\text{c}, L_i)$;
    $\text{Answers}' = evaluateQuery(query(\text{c}'),KB)$;
    if $Answers \cap Answers' = Answers'$ **then** ▷ *i.e.* c' is consistent
        $c = c'$;
return c;

Heuristics Used

Ordering literals

According to arity

According to commonly appearing variables

Domain-specific ordering, according to query type

Usage in reduction step

Literals with higher arity are checked with priority

(e.g., „lake length“ prior to „lake name“, …)

Aims at producing more compact descriptions

Some Examples (Hartfiel)

"Which rivers flow through states which Saarland borders?"

„All that flow through Koblenz." (Mosel, Rhein)

"Which states does the Main flow through?"

„All that border Baden Wuerttemberg and Thueringen"

"Which countries border a state which the Main flows through?"

„All countries that border Bayern." (Austria, Czech Republic)

"Which cities have more than 1500000 inhabitants ?"

„All that are located at the A24 and which a river flows through."

"Which rivers flow through more than 5 cities ?"

„All that flow through Duisburg." (Rhein, Ruhr)

The Situation so far

Some promising results

Limited success

- **not always an intentional description available**

A related field

- **Let us check linguistic approaches**
  - **generating referring expressions**

Generating Referring Expressions (GRE)

Given

A set of objects,

described in terms of entries in a knowledge base

(focusing on semantics, abstracting from surface)

Goal specification

A referring expression that identifies (uniquely)

the intended referent(s) most naturally

Comparison

Exactly the same task (approaches are quite different!)

Basic Ingredients in GRE

Algorithmic issues

Incrementally builds a description out of components

that apply to the intended referent(s)

(attributes, limited uses of relations)

Coverage of referents

Mostly elaborated/used for single referents

Works also for sets of referents with a joint description

Comparison

Orthogonal search organization, same limitation

Some Psychological Insights

Human preferences

Basic categories (*dog* as opposed to *poodle*)

Easy perceivable attributes (color quite prominent)

Redundant expressions (*some* extra attributes)

Personal preferences

Using location attributes or not

More or less redundancy

Salience **is the driving factor**

Search organization

Database context

Increasingly covering positive *items*, one-by-one

Starting with relatively large descriptions

Combining descriptions, eliminating components

Finishing if all positive items are covered

GRE context

Increasingly adding *descriptors/attributes*, one-by-one

Starting with a simple description (e.g., object category)

Adding descriptors, checking termination

An Extension in GRE - for Sets of Objects

*The basic idea (***van Deemter 2000***)*

Boolean combinations **of attributes, not only single attributes**

Search technique proposed

Increasingly complex combinations considered

(single attributes, combinations of two attributes, ...)

An important theoretical result

Identifying description (intentional answer) is guaranteed

(if it exists; it must exist in the database context)

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			•	•			•	•		•		•	

Descriptions with Boolean Combinations

From the perspective of intensional answers to databases

No single answer possible (no set of total commonalities)

Several partially covering answers, varying complexity

At worst only an enumeration, equal to extensional answers

Dealing with near-misses becomes of interest

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

Boolean combinations need transformations (using distributivity)

“the vehicles that are a sport car or small and
either a truck or not red” ->
“the sport cars that are not red and the small trucks”

Recasting Descriptions

Techniques

Partitioning a description by 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)

Extension for Sets of Objects - Critique

Consequences of the (incremental) search technique proposed

Solution quality may be very low

(expressions generated may get very complex/redundant)

Strong commitment:

A priori inclusion of structurally simpler combinations

Turns out to be computationally expensive

Methods proposed

Constraint-Based searching (Gardent 2002)

Best-first Searching (Horacek 2003, …)

Issues of Complexity

	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**

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

Transfer to Database Answers (Ambadi 2018)

Algorithmic Extension of the inductive logic approach

Using boolean extensions incrementally

Complexity may be considerable: 24526 combinations for
"Through which states does Mississippi river flow?"
(11 extensions, takes 249 secs)

Conceptual use

Combining with single extensional answers (near misses)

Additions

Intensional answer *plus* single extensional answer

Exceptions

Intensional answer *except* single extensional answer

Extended Algorithm

Disjunctive descriptions(Extensional answers, KnowledgeBase KB)

for all(Extensional Answers) do
$Ans = ExtensionalAnswers.getNext();$
$C = constructClause(Ans, KB);$ ▷ (first answer clause)
$Ans' = ExtensionalAnswers.getNext();$
$C' = constructClause(Ans', KB);$ ▷ (second answer clause)
Coverage(L_i, V_i)
if Coverage < answers set.size() **then**
  for ((L_i, V_i)in C) ▷ L_i,V_i = literal value pair in first clause
    for((L_j, V_j) in C') ▷ L_j, V_j = literal value pair in first clause
      Query = BoolCombQuery(L_i, V_i, L_j, V_j) ▷ boolean combinations
  check consistency of the combination
  if output set = answers set **then**
return(literal, value) pair; ▷ Intensional answer clause

Some Examples (Ambadi 2018)

“Which state does Missouri river pass through”?

„All the states that border South Dakota except Wyoming“

“Give me the states that border with Arizona”

„Colorado and all states that are located west and through which Colorado r, Truckee rivers flow“

“What states are next to Oregon?”

“All states which either has BorderName as [Arizona, Nevada, Oregon] and landelevationName as [borah peak and snake river] or has Location as [West] and LakeName as [Tahoe]”

Describes Idaho + (California, Nevada)

Future Research

Algorithmic issues

Improving efficiency of the approach pursued

(use of heuristics, technical improvements)

Investigating the referring expressions approach

(completely unexplored yet)

Psychological/coherence issues

Interpreting psychological insights for database answers

Developing relations between query types and

suitable answers components (focus, coherence)

Reusing results for heuristics in the algorithmic part

References

Philipp Cimiano, Sebastian Rudolph, Helena Hartfiel, 2010. Computing intensional answers to questions- An inductive logic programming approach. Data Knowledge Engineering 69, 261-278.

Amihai Motro, 1994. Intensional Answers to Database Queries.

S.-C. Yoon, I.-Y. Song, E.K. Park, 1994. Intelligent query answering in deductive and object-oriented databases, in: Proceedings of the Third International Conference on Information and Knowledge Management (CIKM), pp. 244-251.

Farah Benamara, 2004. Generating Intensional Answers in Intelligent Question Answering, Institut de Recherches en Informatique de Toulouse IRIT;France Systems.

Kees van Deemter, (2002). Generating referring expressions: Boolean extensions of the incremental algorithm. Computational Linguistics, 28:3752.

Helmut Horacek, 2003. A Best-First Search Algorithm for Generating Referring Expressions, Universität des Saarlandes, Germany.

Jette Viethen, Robert Dale, 2006. Algorithms for Generating Referring Expressions: Do They Do What People Do? COLING/ACL 2006: INLG-06 - 4th International Natural Language Generation Conference, Proceedings of the Conference.

Claire Gardent, 2002. Generating Minimal Definite Descriptions. In Proc. of ACL-2002, pp. 96-103.

Sampath Ambadi, 2018. Intensional Answers to Database Queries, Master thesis, Saarland University.