Graph-Based Question Answering

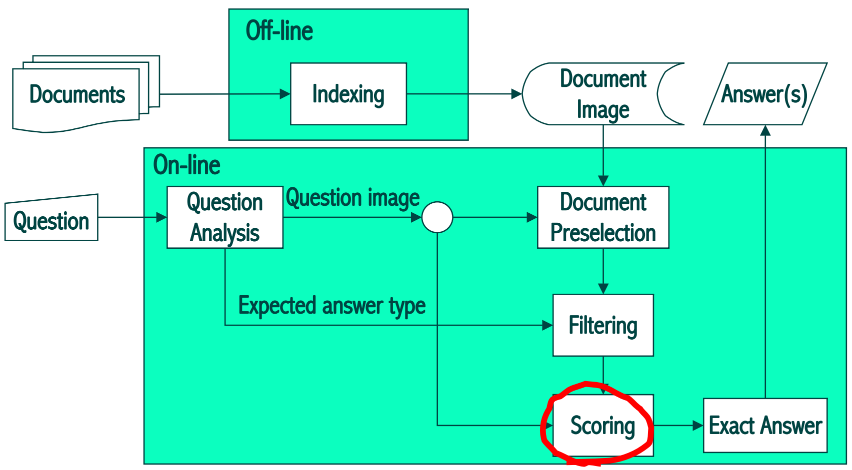
Diego Mollá-Aliod

30 August 2004

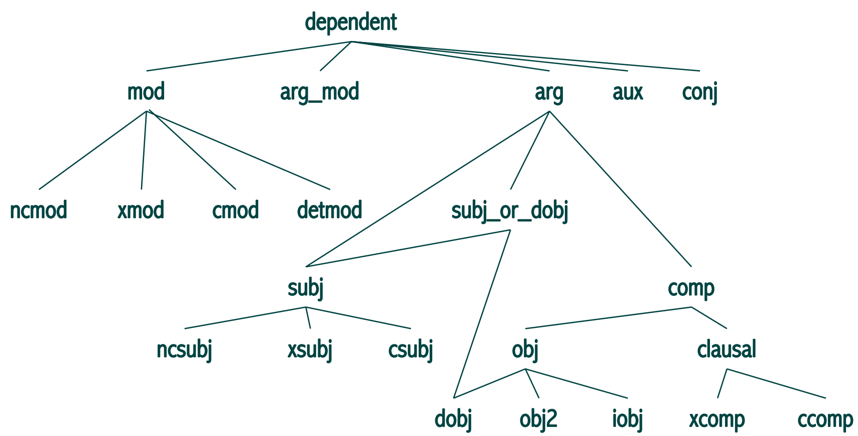
Outline

- Question Answering and AnswerFinder
- Conceptual Graphs
- Graph Comparison

Architecture of AnswerFinder



Grammatical Relations



Grammatical Relations

– *A man named Richard Sears has been playing a joke on shoppers.*
(detmod _ man a)
(subj name man _) (dobj name richard_sears _)
(detmod _ joke a) (subj play man _) (aux _ play have)
(aux _ play be)
(ncmod shopper play on) (dobj play joke _)

– *Who played a joke on shoppers?*
(subj play who _) (dobj play joke _)
(ncmod shopper play on) (detmod _ joke a)

Minimal Logical Forms

- Called Minimal Logical forms because they encode the minimum information required for AE
- Flat expressions that use reification
- Example: *cp will quickly copy files*
  holds(e4), object(cp,o1,[x1]), object(s_command,o2,[x1]), evt(s_copy,e4,[x1,x6]), object(s_file,o3,[x6]), prop(quickly,p3,[e4]).
- Example: *the man that came ate bananas and apples with a fork*
  holds(e1), object(s_man,o2,[x2]), evt(s_come,e4,[x2]), evt(s_eat,e5,[x7]), x6@<x7, x8@<x7, object(s_banana,o6,[x6]), object(s_apple,o8,[x8]), prop(with,p9,[e5,x11]), object(s_fork,o11,[x11]).

Minimal Logical Forms

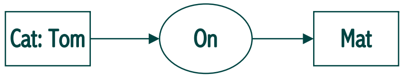
– *A man named Richard Sears has been playing a joke on shoppers.*
holds(o10), object('man',o2,[x2]),
evt('name',e3,[X3,x2,x4]),
object('richard_sears',o4,[x4]), evt('play',e8,[x2,x10]),
object('joke',o10,[x10]), prop('on',p11,[e8,x12]),
object('shopper',o12,[x12])

– *Who played a joke on shoppers?*
holds(e2), object('who',o1,[x1]), evt('play',e2,[x1,x4]),
object('joke',o4,[x4]), prop('on',p5,[e2,x6]),
object('shopper',o6,[x6])

Outline

- Question Answering and AnswerFinder
- Conceptual Graphs
- Graph Comparison

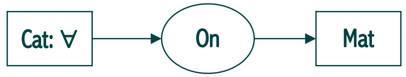
Conceptual Graphs



- Concept Node [Cat:Tom], [Mat]
  - Entities, attributes, or events (actions)
  - Concept nodes have two attributes:
    - Type (e.g. Cat)
    - Referent (e.g. Tom)
- Relation Node (On)
  - The kind of relationship between two concept nodes

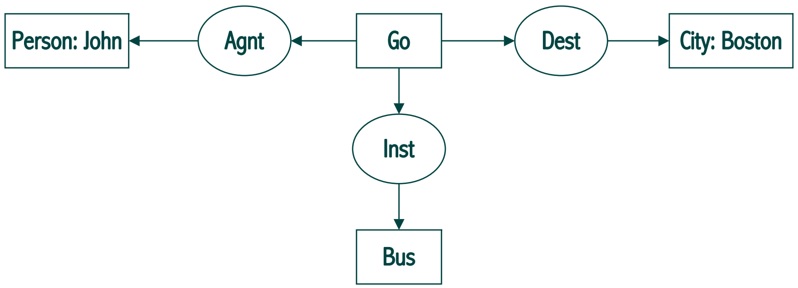
Quantification

- *Every cat is on a mat*



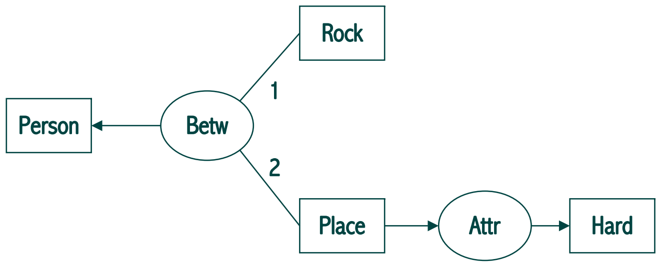
Thematic Roles

- *John is going to Boston by Bus*



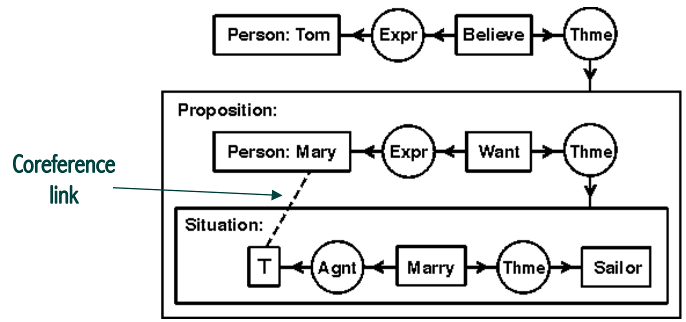
N-Ary Relations

- *A person is between a rock and a hard place*



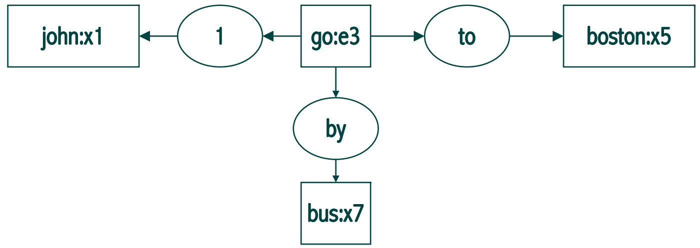
Nested Conceptual Graphs

- *Tom believes that Mary wants to marry a sailor*



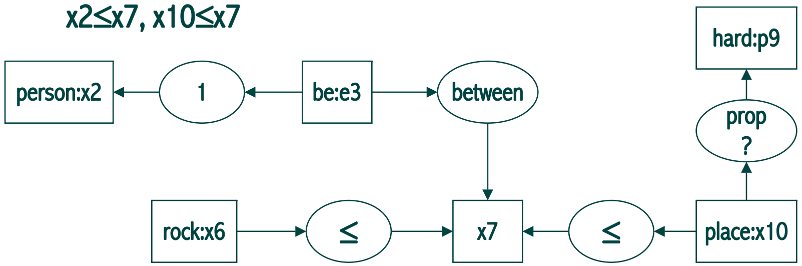
How does it Compare with AnswerFinder?

- *John is going to Boston by bus*
  holds(e3), prop('by',p6,[e3,x7]), prop('to',p4,[e3,x5]), object('john',o1,[x1]), evt('go',e3,[x1]), object('bus',o7,[x7]), object('boston',o5,[x5])



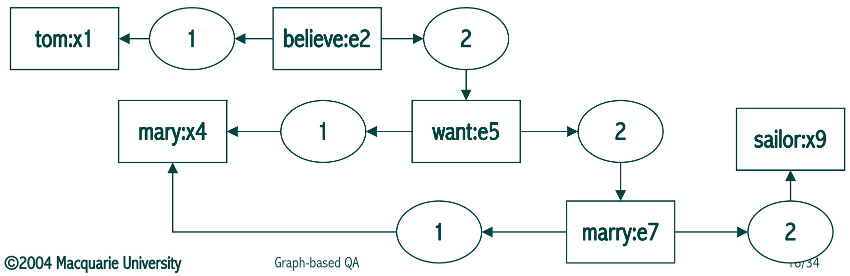
How does it Compare with AnswerFinder?

- *A person is between a rock and a hard place*
  holds(e3), object('rock',o6,[x6]), evt('be',e3,[x2]), prop('hard',p9,[x10]), object('person',o2,[x2]), prop('between',p4,[e3,x7]), object('place',o10,[x10]), x2≤x7, x10≤x7



How does it Compare with AnswerFinder?

- *Tom believes that Mary wants to marry a sailor*
  holds(e2), evt('marry',e7,[x4,x9]), object('sailor',o9,[x9]), evt('believe',e2,[x1,e5]), object('tom',o1,[x1]), object('mary',o4,[x4]), evt('want',e5,[x4,e7])

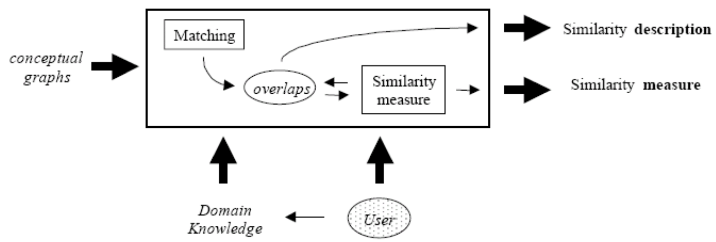


Outline

- Question Answering and AnswerFinder
- Conceptual Graphs
- Graph Comparison

Comparison of Conceptual Graphs

- Two steps:
  1. Find an overlap
     – Use domain knowledge: thesauri and *isa* hierarchies
  2. Compute the similarity in function of the overlap



Overlap – The Intuition

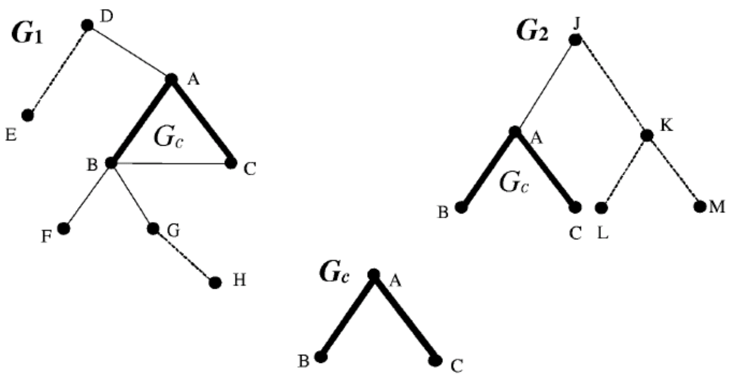


Fig.1. Intersection of two conceptual graphs

Dice Coefficient

$$S_{D_1,D_2} = \frac{2n(D_1 \cap D_2)}{n(D_1) + n(D_2)}$$

- $n(D_i)$ = number of terms in D_i
- $n(D_i \cap D_j)$ = number of terms that D_i and D_j have in common

Applying the Dice Coefficient

- Conceptual Similarity
  - n(G) is the number of concept nodes of graph G

$$S_c = \frac{2n(G_c)}{n(G_1) + n(G_2)}$$

- Relational Similarity
  - m(G) is the number of relations of graph G
  - $m_{G1}(G2)$ is the number of relations in the immediate neighbourhood of G1 in G2, where G1 is a subgraph of G2

$$S_r = \frac{2m(G_c)}{m_{G_c}(G_1) + m_{G_c}(G_2)}$$

Calculation of Relational Similarity

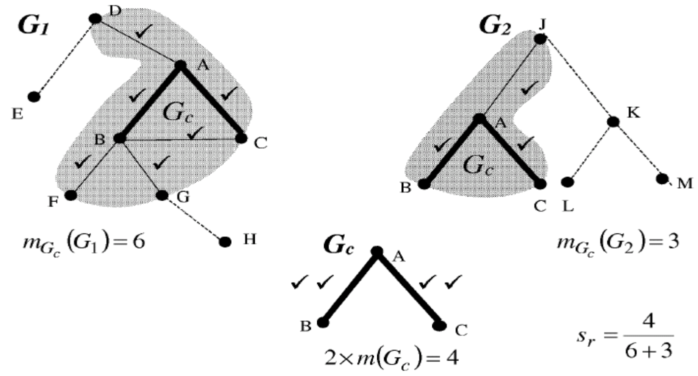


Fig. 2. Calculation of relational similarity

Generalisation of a Conceptual Graph

- Unrestrict rule:
  - Replace the type label of a concept with a supertype

  or

  - Replace an individual referent with a generic one
- Detach rule (??):
  - Split a node into two with the same type and referent

  and

  - Distribute the relations of the original node between the two resulting nodes

Projection

- v is a generalisation of u $(u \le v)$
- we can define a projection $\pi: v \rightarrow u$

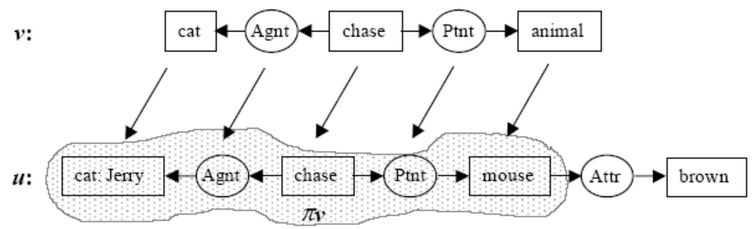


Fig. 2. Projection mapping π: $v \rightarrow u$ (the highlighted area is the projection of v in u).

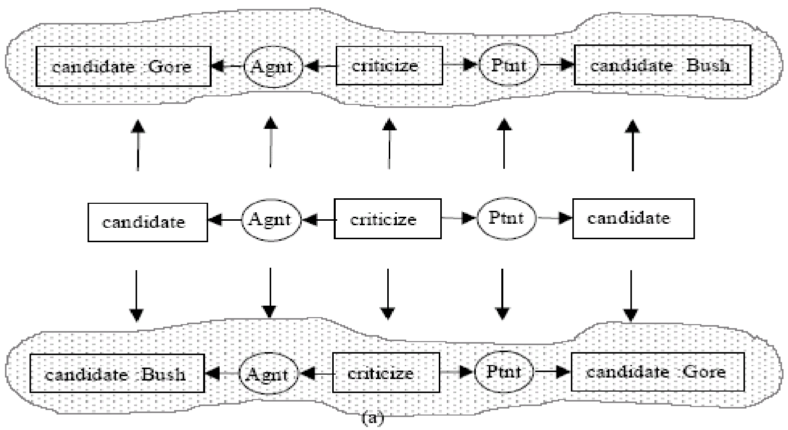
Overlap with Graph Generalisations

- v is a <u>common generalisation</u> of u_1 and u_2 iff $u_1 \leq v$ and $u_2 \leq v$
- A set of common generalisations of u_1 and u_2 is <u>compatible</u> iff they have projection maps such that the corresponding projections in G, u_1 and u_2, do not intersect
- A compatible set of common generalisations $\{g_1 \ldots g_n\}$ of u_1 and u_2 is <u>maximal</u> iff we cannot add a new common generalisation $g \leq g_i$ such that $\{g_1 \ldots g_n, g\}$ is compatible
- A set of common generalisations of u_1 and u_2 is an <u>overlap</u> iff it is compatible and maximal

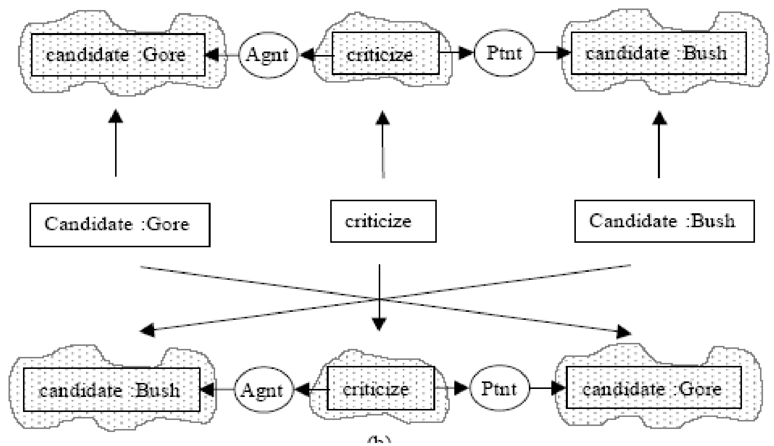
Finding an Overlap

- There may be several overlaps
- Finding an overlap is NP-complete
- Still, workable for small graphs

One Overlap



Another Overlap



Conceptual Similarity

$$S_c = \frac{2 \times \sum_{c \in \bigcup O} (weight(c) \times \beta(\pi_{G_1} c, \pi_{G_2} c))}{\sum_{c \in G_1} weight(c) + \sum_{c \in G_2} weight(c)}$$

$$weight(c) = \begin{cases} W_E & \text{if } c \text{ represents an entity} \\ W_V & \text{if } c \text{ represents an action} \\ W_A & \text{if } c \text{ represents an attribute} \end{cases}$$

$$\beta(\pi_{G_1} c, \pi_{G_2} c) = \begin{cases} 1 & \text{if } type(\pi_{G_1} c) = type(\pi_{G_2} c) \text{ and } referent(\pi_{G_1} c) = referent(\pi_{G_2} c) \\ \frac{depth}{depth + 1} & \text{if } type(\pi_{G_1} c) = type(\pi_{G_2} c) \text{ and } referent(\pi_{G_1} c) \neq referent(\pi_{G_2} c) \\ \frac{2 d_c}{d_{\pi_{G_1} c} + d_{\pi_{G_2} c}} & \text{if } type(\pi_{G_1} c) \neq type(\pi_{G_2} c) \end{cases}$$

Relational Similarity

$$S_r = \frac{2 \times \sum_{r \in \bigcup O} weight_O(r)}{\sum_{r \in N_O(G_1)} weight_{G_1}(r) + \sum_{r \in N_O(G_2)} weight_{G_2}(r)}$$

$$N_O(G_i) = \bigcup_{c \in O} N_{G_i}(\pi_{G_i} c), \text{ where } \mathrm{N_G}(c) = \{r \mid r \text{ is connected to } c \text{ in } G\}$$

$$weight_G(r) = \frac{\sum_{c \in N_G(r)} weight(c)}{|N_G(r)|}, \text{ where } N_G(r) = \{c \mid c \text{ is connected to } r \text{ in } G\}$$

The Similarity Measure

$$s = s_c \times (a + b \times s_r)$$

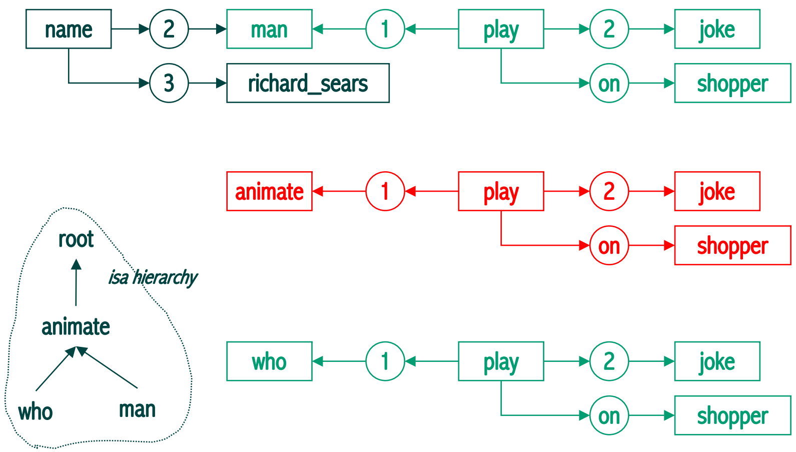
- The coefficients *a* and *b* reflect user-specified balance

  $0 < a, b < 1$ and $a + b = 1$

Example

Conditions	Overlap	s_c	s_r	s
$a = 0.1$, $\boldsymbol{b = 0.9}$ $w_E = w_V = w_A = 1$	[candidate]←(agt)←[criticize]→(pnt)→[candidate]	0.86	1	0.86
	[candidate:Bush] [criticize] [candidate:Gore]	1.00	0	0.10
$\boldsymbol{a = 0.9}$, $b = 0.1$ $w_E = w_V = w_A = 1$	[candidate]←(agt)←[criticize]→(pnt)→[candidate]	0.86	1	0.86
	[candidate:Bush] [criticize] [candidate:Gore]	1.00	0	0.90
$a = 0.5$, $b = 0.5$ $\boldsymbol{w_E = 2}$ $w_V = w_A = 1$	[candidate]←(agt)←[criticize]→(pnt)→[candidate]	0.84	1	0.84
	[candidate:Bush] [criticize] [candidate:Gore]	1.00	0	0.50

Example with AnswerFinder



The Similarity

- $W_E = W_V = W_A = 1$:

$$S_c = \frac{2\times\left(\frac{2\times 1}{2+2}+1+1+1\right)}{6+4} = 0.7 \qquad S_r = \frac{2\times\left(\frac{2}{2}+\frac{2}{2}+\frac{2}{2}\right)}{\left(\frac{2}{2}+\frac{2}{2}+\frac{2}{2}+\frac{2}{2}\right)+\left(\frac{2}{2}+\frac{2}{2}+\frac{2}{2}\right)} = 0.857$$

  - $a = b = 0.5$:

    $$S = 0.7\times(0.5+0.5\times 0.857) = 0.65$$

  - $a = 0.9$, $b = 0.1$:

    $$S = 0.7\times(0.9+0.1\times 0.857) = 0.69$$

  - $a = 0.1$, $b = 0.9$:

    $$S = 0.7\times(0.1+0.9\times 0.857) = 0.61$$