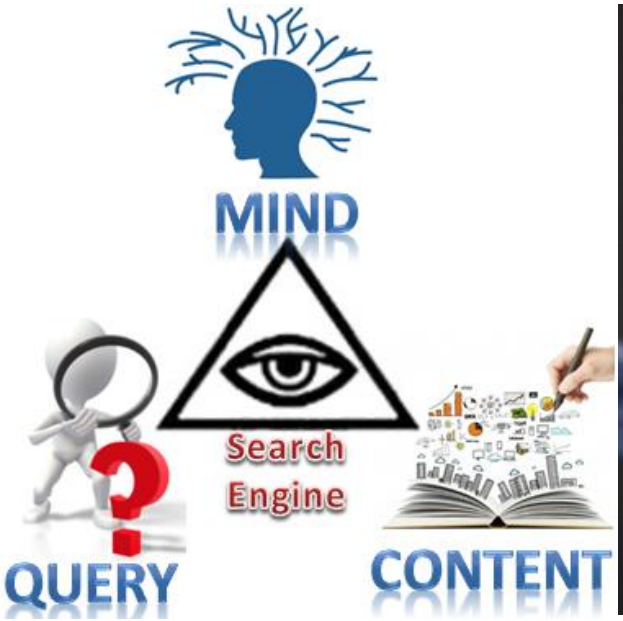






TB-Structure: Collective Intelligence for Exploratory Keyword Search

Vagan Terziyan, Mariia Golovianko & Michael Cochez

IKC-2016, Cluj-Napoca, Romania, 8-9 September 2016

Check updates here: http://www.mit.jyu.fi/ai/IKC-2016.pptx

keystone

The Authors



Vagan Terziyan, Professor (Distributed Systems), Faculty of Information Technology, University of Jyväskylä (**FINLAND**),

e-mail:
vagan.terziyan@jyu.fi .



Mariia Golovianko, PhD, Department of Artificial Intelligence, Kharkiv National University of Radioelectronics (**UKRAINE**),

ACKNOWLEDGEMENT: this research has been supported by the STSM grant from KEYSTONE (COST ACTION IC1302)

e-mail:
mariia.golovianko@nure.ua ;
golovianko@gmail.com .

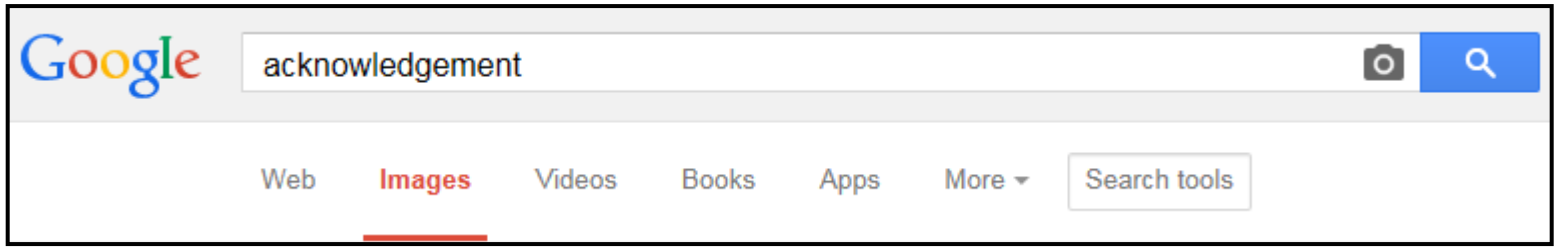


Michael Cochez, PhD, University of Jyväskylä (**FINLAND**), Currently: postdoctoral researcher at the Fraunhofer Institute for Applied Information Technology FIT / RWTH University in Aachen (**GERMANY**)

e-mail:
michael.cochez@jyu.fi ;
michael.cochez@fit.fraunhofer.de .



- We are grateful to the anonymous photographers and artists, whose photos and pictures (or their fragments) posted on the Internet, we used in the presentation.





Beware on hidden slides!

"Exploratory searcher has a set of search criteria in mind, but does not know how many results will match those criteria — or if there even are any matching results to be found" (Tunkelang, 2013)

Exploratory Search



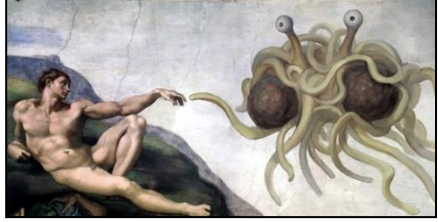
Exploratory search covers a broader class of information exploration activities than typical information retrieval and these activities are usually carried out by searchers who are, according to White and Roth (2009):

- unfamiliar with the domain of their search objective, i.e., unsure how to formulate their objective;
- or unsure about the ways (technology or process) to approach their objective;
- or unsure about their search objectives in the first place.

Typically, therefore, such searchers generally combine querying and browsing strategies to foster learning and investigation.

An example scenario, often used to motivate the research by mSpace (http://mspace.fm/), states: "if a user does not know much about classical music, how should they even begin to find a piece that they might like".

The Open World Assumption (Interpretations)



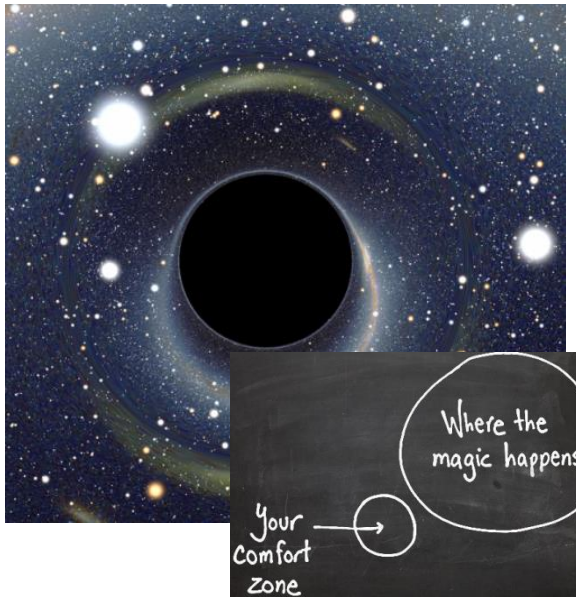
- "*Students need to be prepared for jobs that do not yet exist ... using technologies that have not yet been invented ... in order to solve problems that we do not even know are problems yet*".
  – [Richard Riley, Secretary of Education under Clinton]



The Open World Assumption (OWA): a lack of information does not imply the missing information to be false.

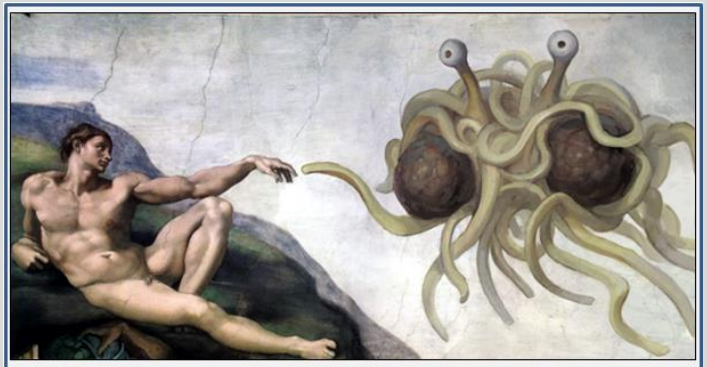
Knowledge is never complete — gaining and using knowledge is a permanent evolutionary process, and is never complete. A completeness assumption around knowledge is by definition inappropriate;

- *Search algorithms need to be prepared for content instances that do not yet visible or exist ... which may have keywords that have not yet been invented or cannot yet be formulated ... in order to get meaningful search outcome to be used for the problems that we do not even recognize to be our problems yet.*



Closed World Information Retrieval vs. Exploratory Search based on the Open World Assumption







Open World Assumption (OWA):
Knowledge is never complete, it evolves together with its content, context, structure and sources …

Search Query

Q_i

OWA-Driven Iterative Discovery Engine

Q_{i+1}

Discovered update on Search Query

Discovered Content

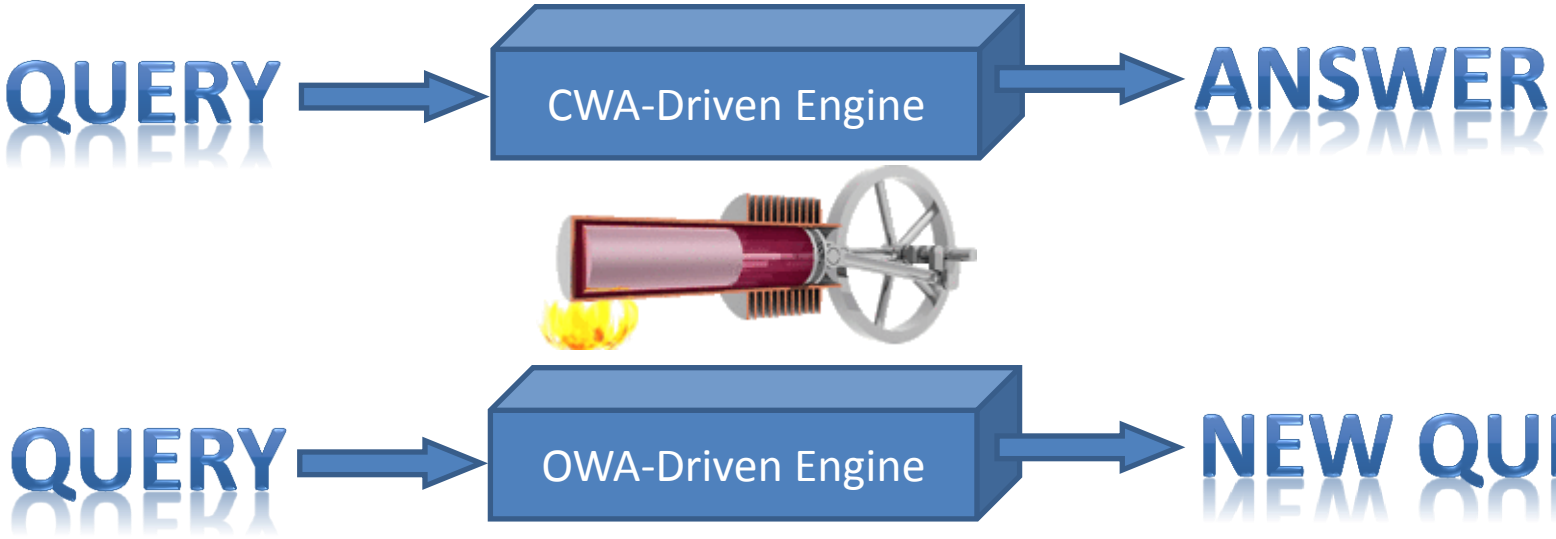
OUT_i

Data Mining and Query Refinement

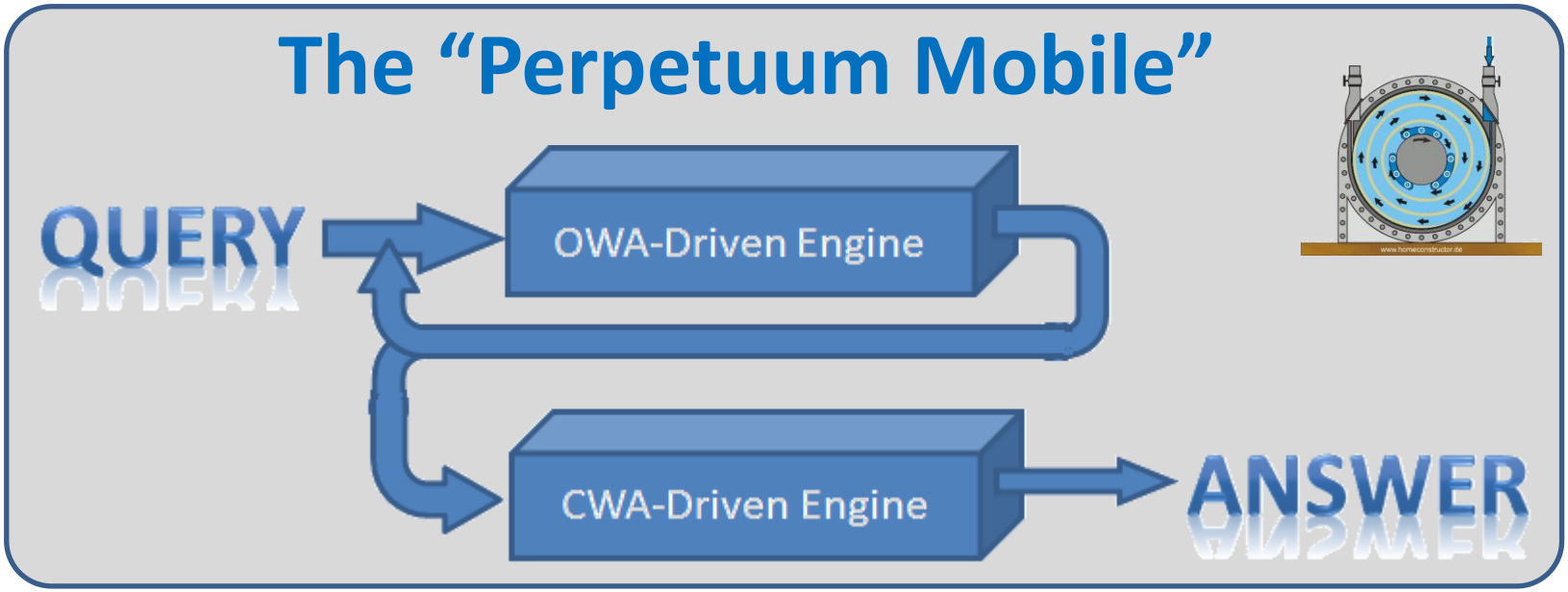
Generated "query trail":
$\{Q_i \rightarrow Q_{i+1} \rightarrow Q_{i+2} \rightarrow \ldots \rightarrow Q_{i+n}\}$

With the OWA-Driven Search you may discover interesting content from the Web (as well as a promising business opportunity) having no idea in advance what you are searching for !

Information Retrieval (CWA) vs. Exploratory Search (OWA)

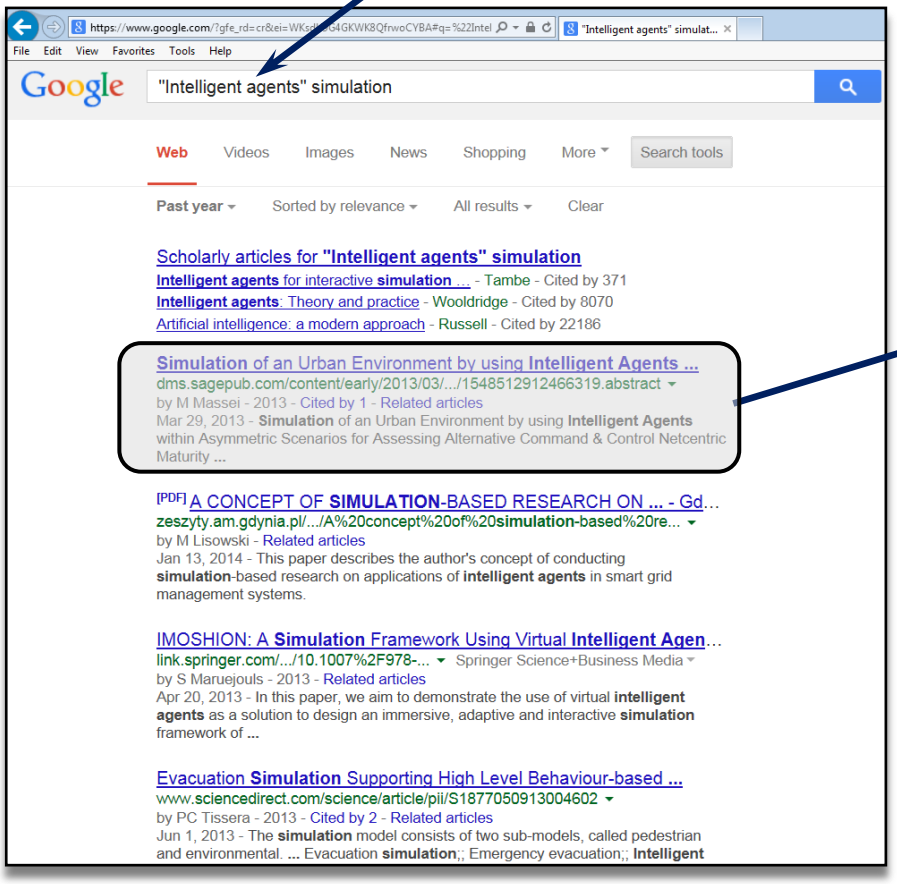


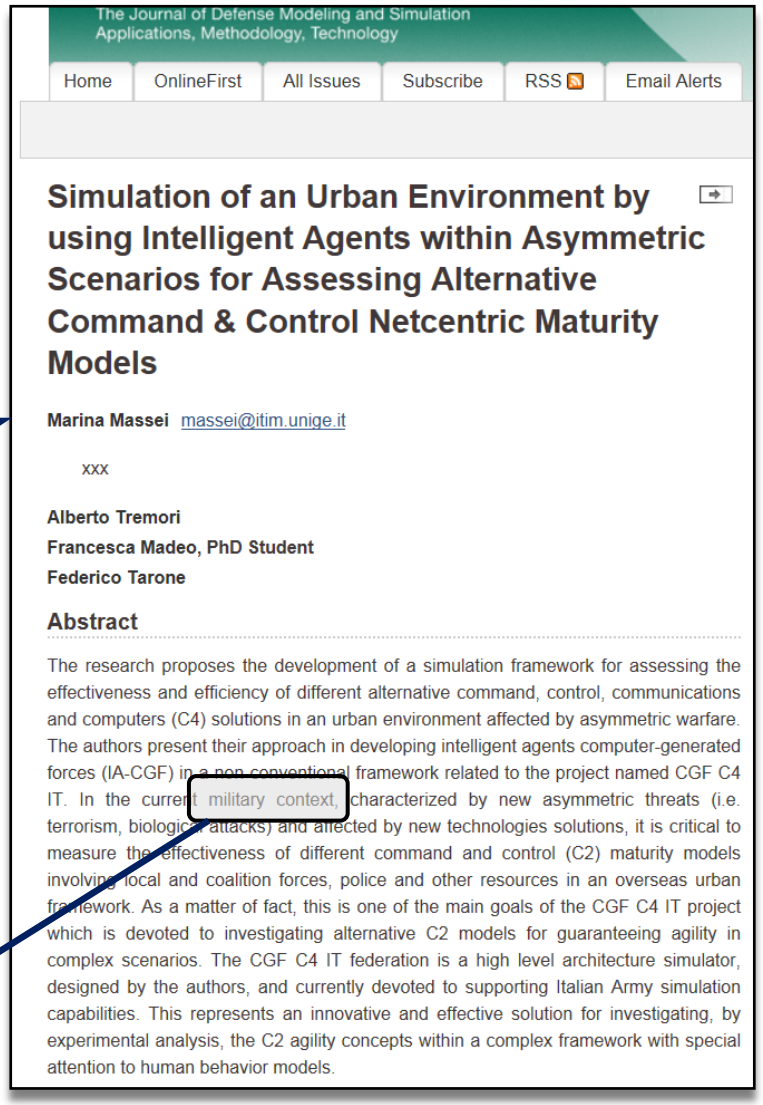
The "Perpetuum Mobile"



Exploratory Search Example (1)

Q_0: { intelligent-agents ; simulation }

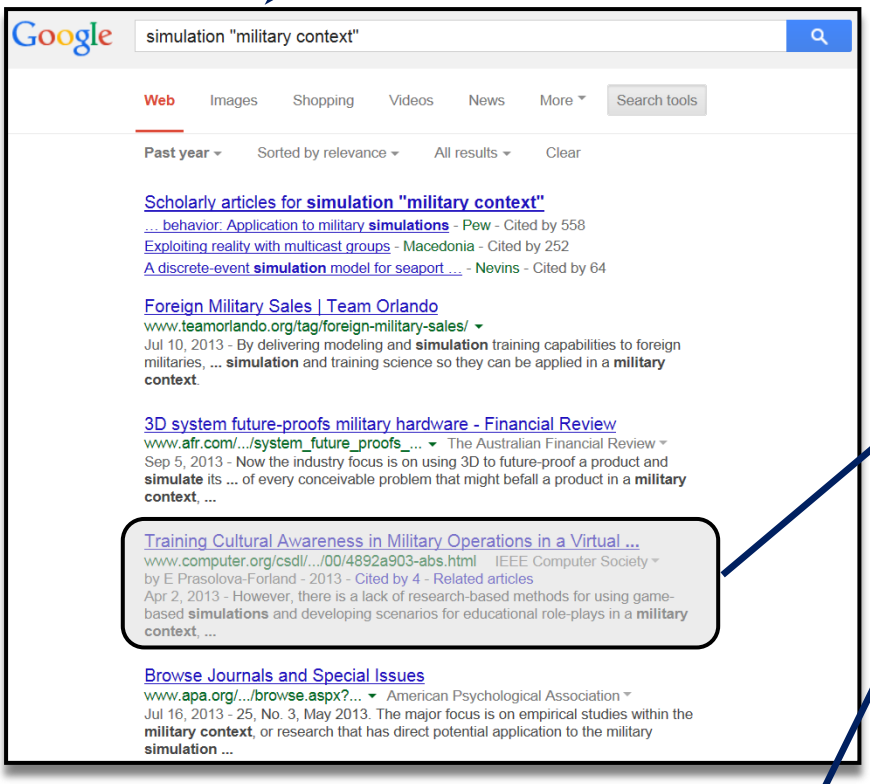




Q_1: {simulation ; military-context }

Exploratory Search Example (2)

Q_1: {simulation ; military-context }

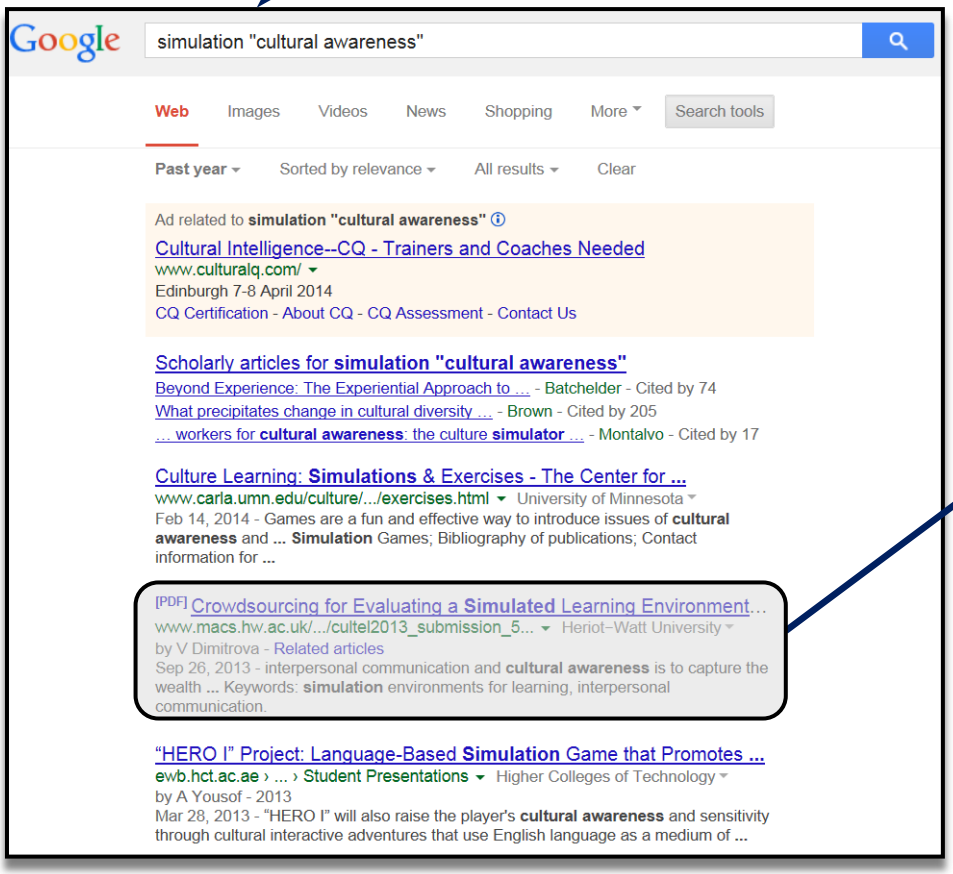


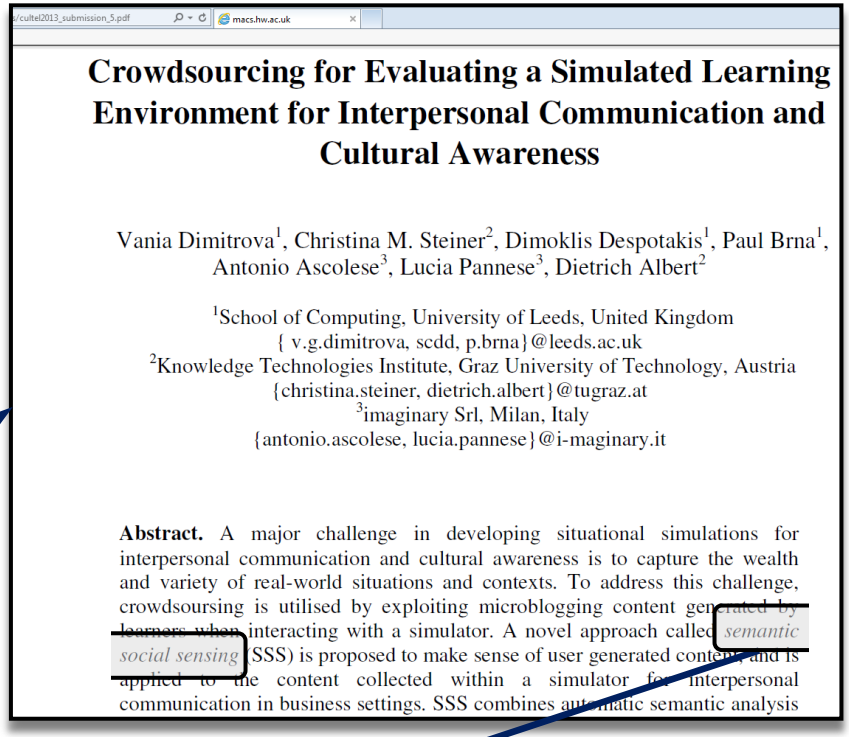


Q_2: { simulation ; cultural-awareness}

Exploratory Search Example (3)

Q_2: { simulation ; cultural-awareness}





Crowdsourcing for Evaluating a Simulated Learning Environment for Interpersonal Communication and Cultural Awareness

Vania Dimitrova[1], Christina M. Steiner[2], Dimoklis Despotakis[1], Paul Brna[1], Antonio Ascolese[3], Lucia Pannese[3], Dietrich Albert[2]

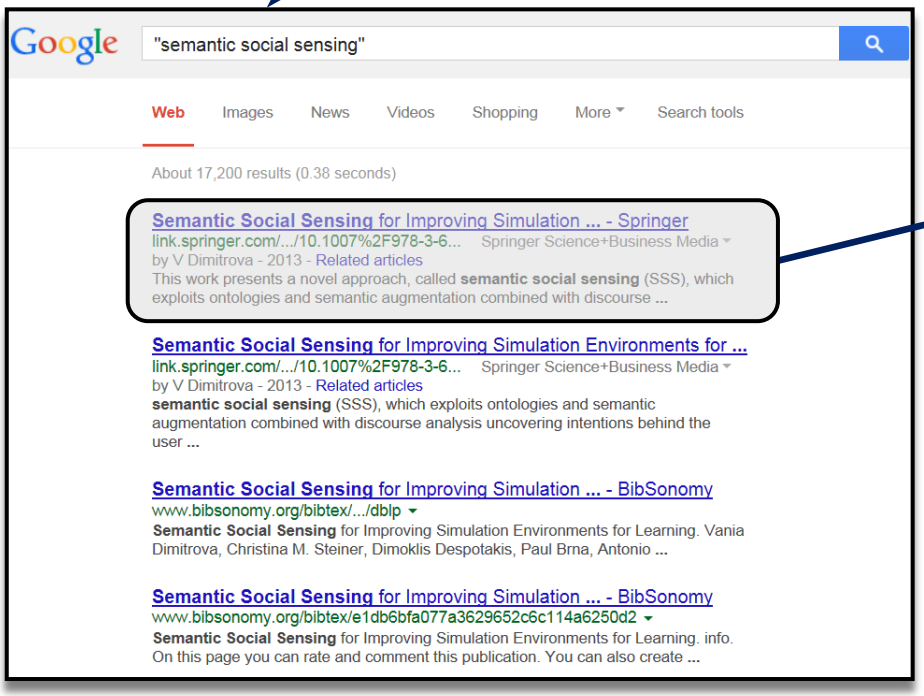
[1]School of Computing, University of Leeds, United Kingdom
{ v.g.dimitrova, scdd, p.brna}@leeds.ac.uk
[2]Knowledge Technologies Institute, Graz University of Technology, Austria
{christina.steiner, dietrich.albert}@tugraz.at
[3]imaginary Srl, Milan, Italy
{antonio.ascolese, lucia.pannese}@i-maginary.it

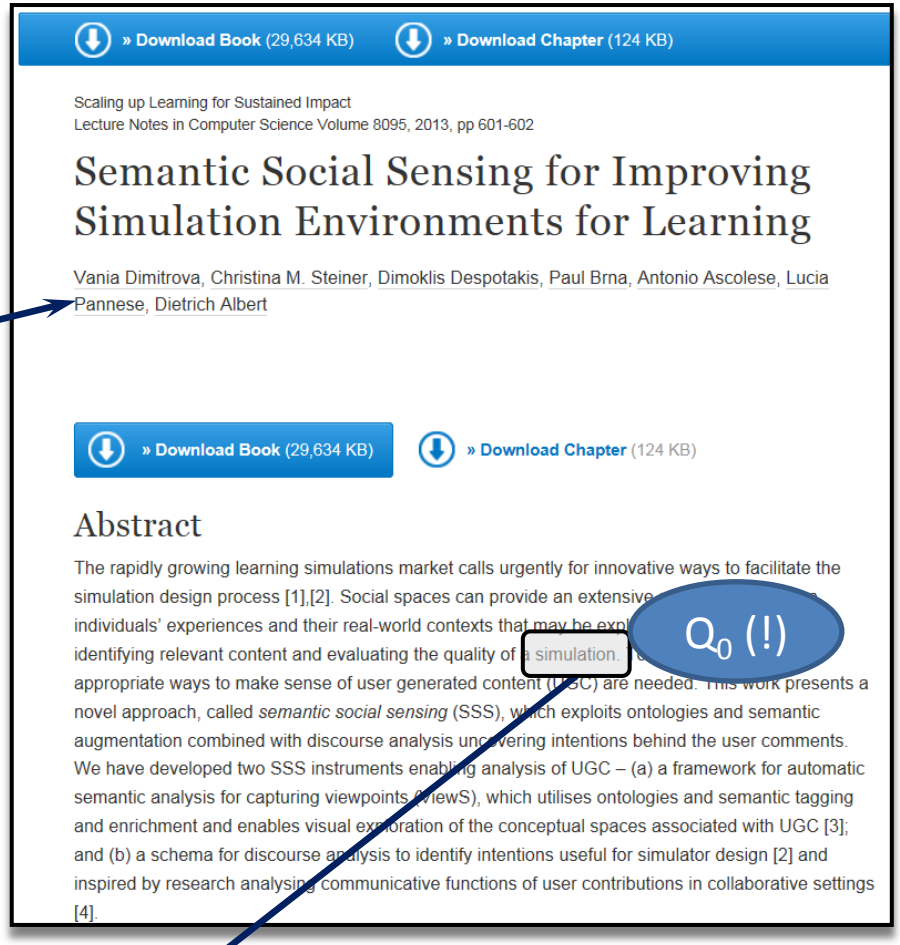
Abstract. A major challenge in developing situational simulations for interpersonal communication and cultural awareness is to capture the wealth and variety of real-world situations and contexts. To address this challenge, crowdsoursing is utilised by exploiting microblogging content generated by learners when interacting with a simulator. A novel approach called *semantic social sensing* (SSS) is proposed to make sense of user generated content and is applied to the content collected within a simulator for interpersonal communication in business settings. SSS combines automatic semantic analysis

Q_3: {semantic-social-sensing }

Exploratory Search Example (4)

Q_3: {semantic-social-sensing }





Q_0 (!)

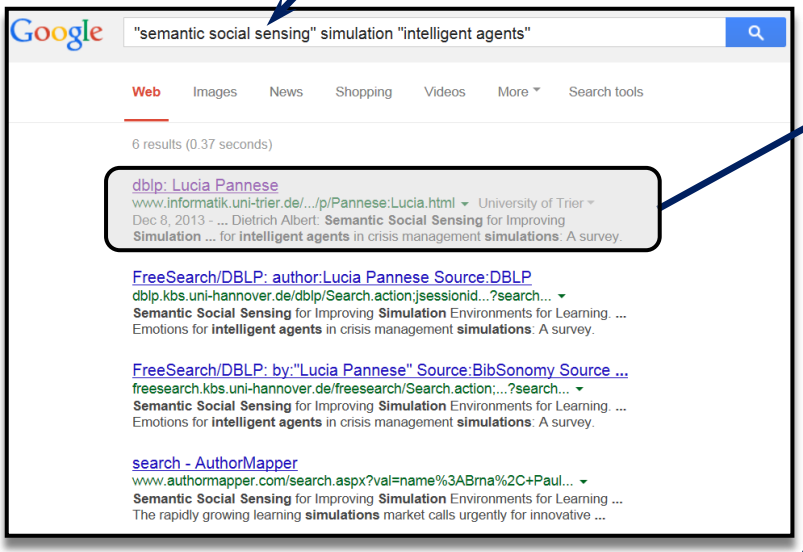
Q_4: {semantic-social-sensing ; simulation ; intelligent-agents}

Intelligent-agents

Q_0 (?)

Exploratory Search Example (5)

Q_4: {semantic-social-sensing ; simulation ; intelligent-agents}

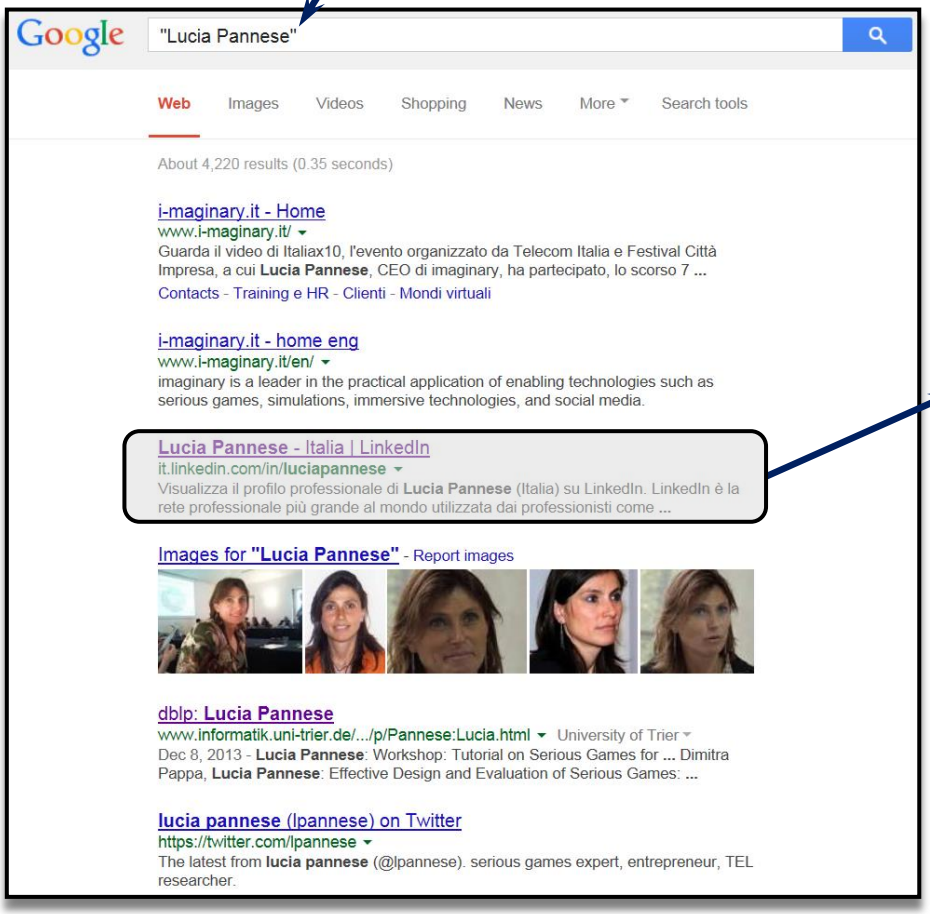


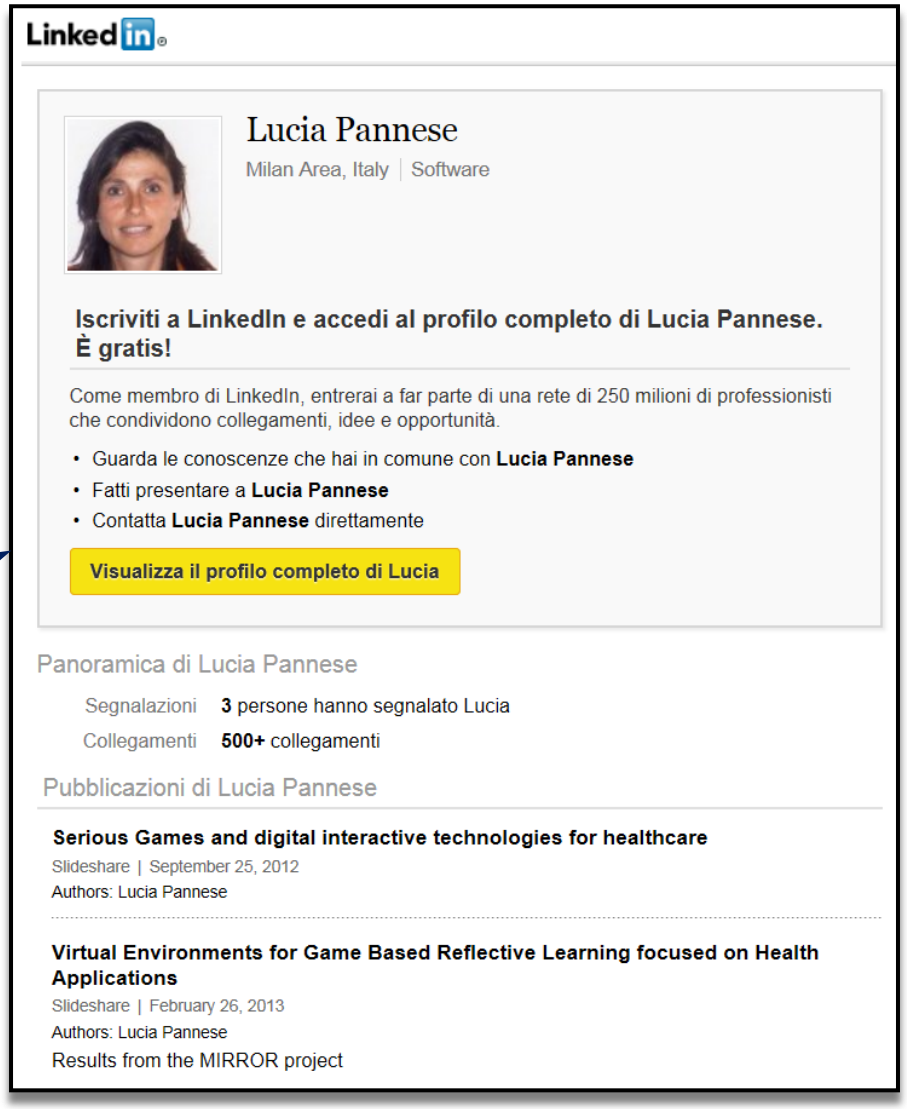


Q_5: { Lucia-Pannese }

Exploratory Search Example (6)

Q_5: { Lucia-Pannese }





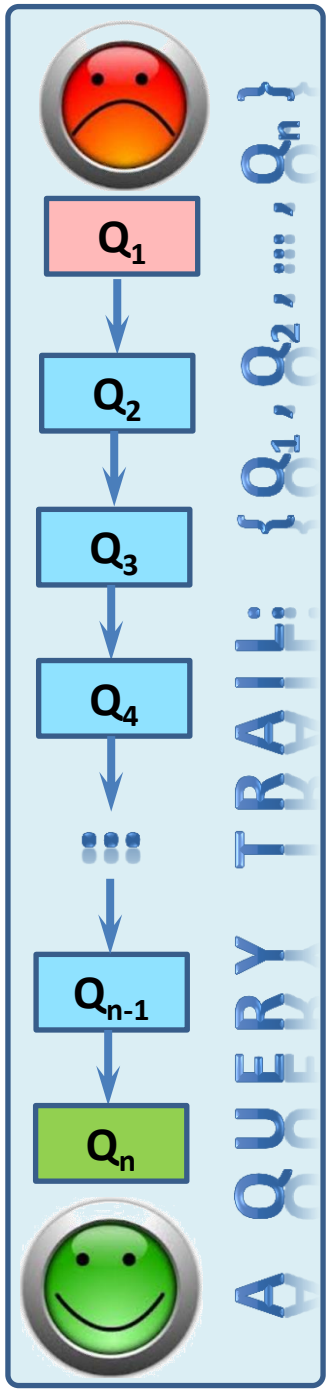
Original query: Q_0: { intelligent-agents ; simulation }

Discovered: new collaboration opportunity "**Lucia Pannese**" !

Query trail: {$Q_0 \rightarrow Q_1 \rightarrow Q_2 \rightarrow Q_3 \rightarrow Q_4 \rightarrow Q_5$ }

Discovered: potentially interesting domain – "**semantic social sensing**" !

Query trails aka “collective intelligence”





Collected query trails:

$\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
$\{Q_{11}, Q_3, Q_9\}$





Collected query trails:

$\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
$\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$





Collected query trails:

$\{Q_1, Q_2, Q_3, Q_9\}$
$\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$



RQ: CAN WE MAKE ANY SENSE FROM SUCH COLLECTION?

Query (Prefix) Tree / Forest:

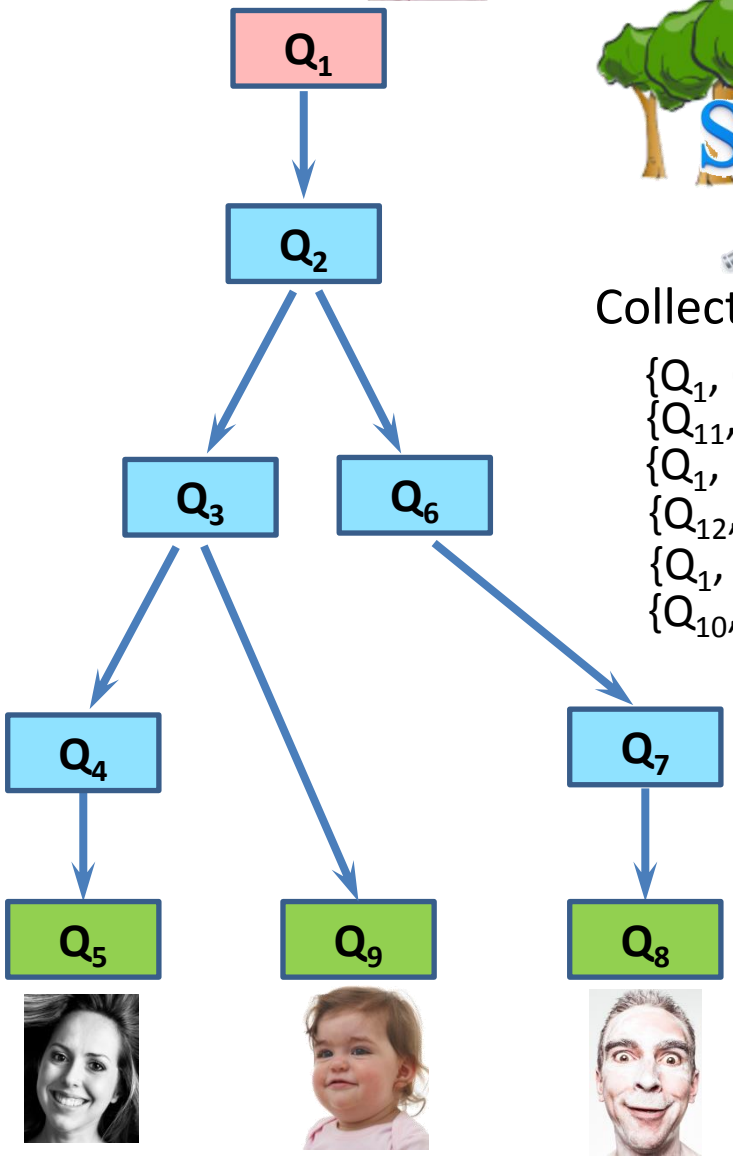
"collective confusion" – "individual satisfaction"

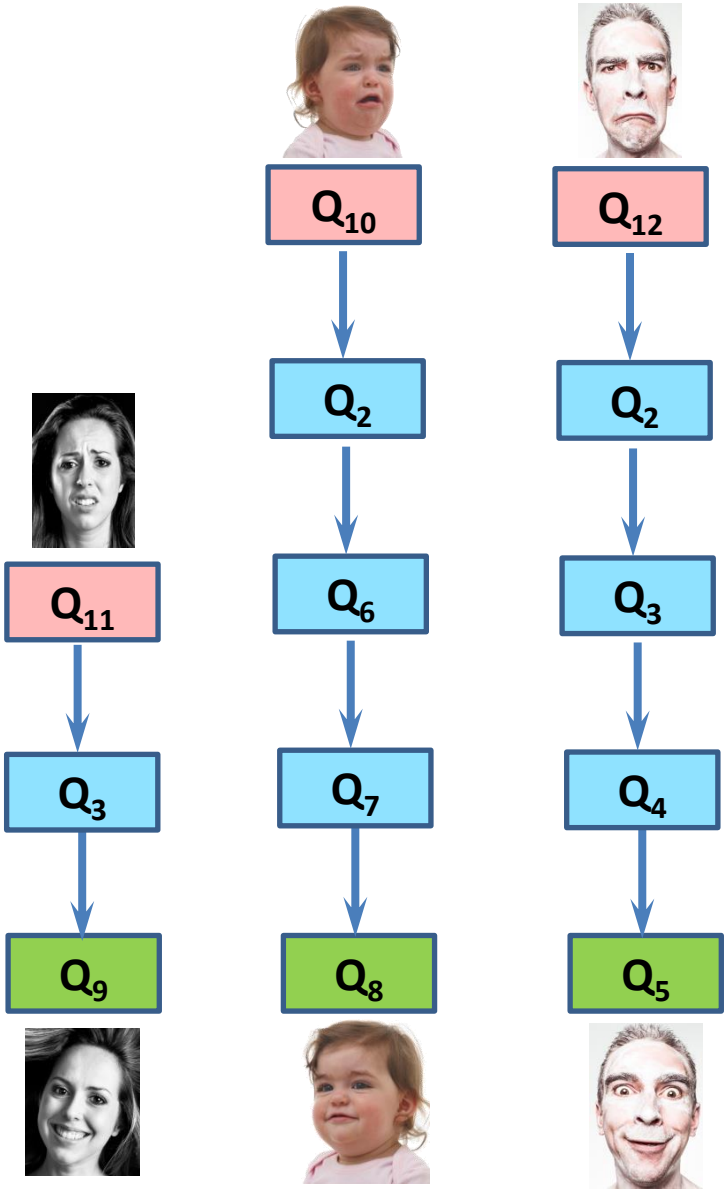




Collected query trails:

$\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
$\{Q_{11}, Q_3, Q_9\}$
$\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
$\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
$\{Q_1, Q_2, Q_3, Q_9\}$
$\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$





Inverted Query (Suffix) Tree / Forest:

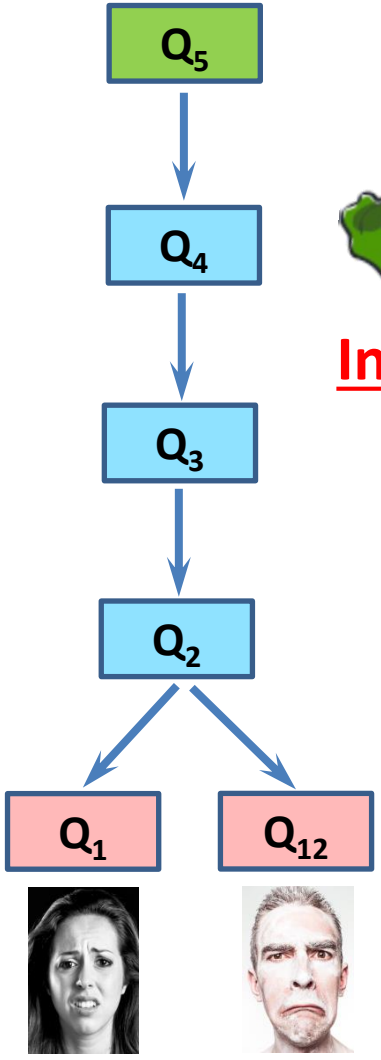
"collective satisfaction" – "individual confusion"

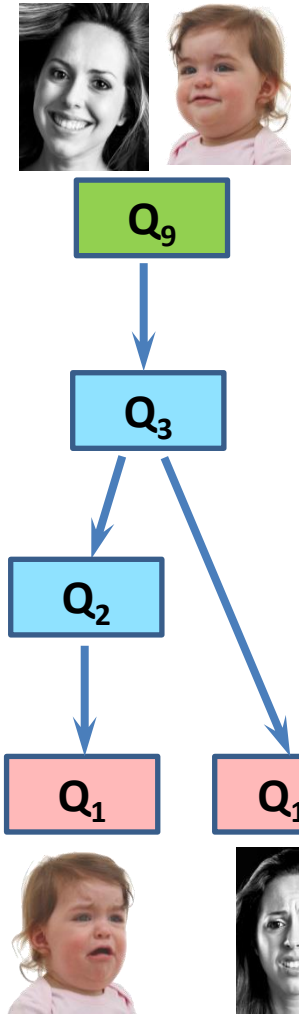


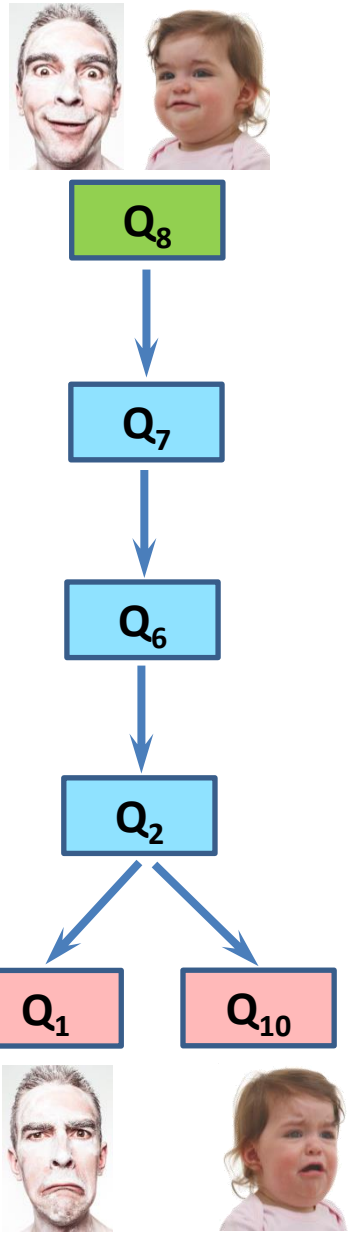


Inverted (!) query trails:

$\{Q_5, Q_4, Q_3, Q_2, Q_1\}$
$\{Q_9, Q_3, Q_{11}\}$
$\{Q_8, Q_7, Q_6, Q_2, Q_1\}$
$\{Q_5, Q_4, Q_3, Q_2, Q_{12}\}$
$\{Q_9, Q_3, Q_2, Q_1\}$
$\{Q_8, Q_7, Q_6, Q_2, Q_{10}\}$







Merged Query Forest with Inversed Query Forest = = There-and-Back-Structure (TB-Query-Structure)



Lovitskii, V. A., Terziyan, V. (**1981**). **Words' Coding in TB-Structure**. *Problemy Bioniki*, 26, 60-68. (*In Russian*)

CELEBRATING 35 YEARS

TB-Query has been originally invented for “intelligent” storage of words

CELEBRATING 35 YEARS

Words:

Ч-И-Т-А-Т-Ь

Ч-И-Т-А-Т-Е-Л-Ь

Ч-И-Т-А-Л

П-И-С-А-Т-Ь

Inverted Words:

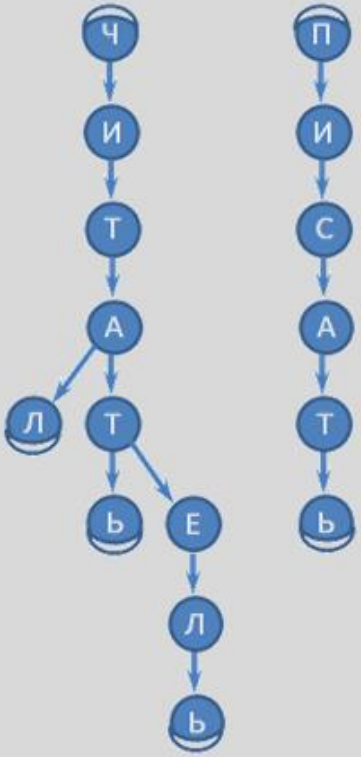
Ь-Т-А-Т-И-Ч

Ь-Л-Е-Т-А-Т-И-Ч

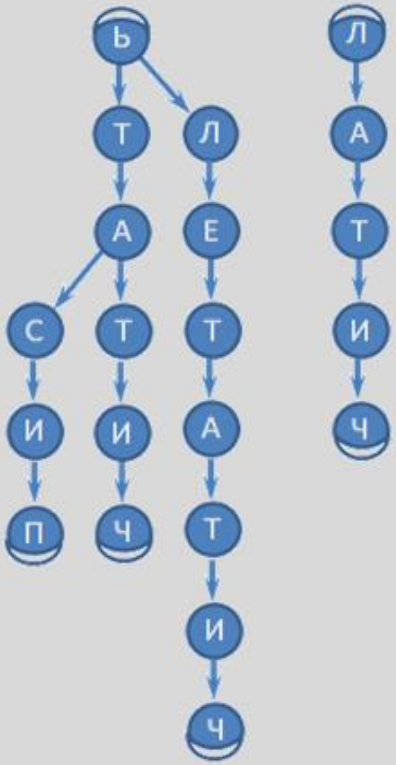
Л-А-Т-И-Ч

Ь-Т-А-С-И-П

Prefix Forest:

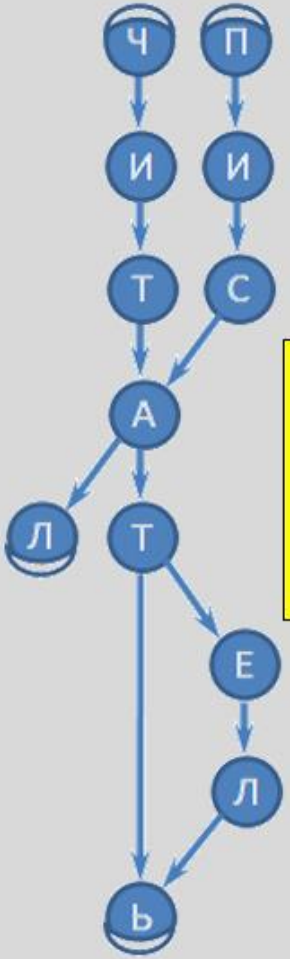


Suffix Forest:



TB-Structure:

Merged Prefix & Suffix Forests



New words appeared (!):

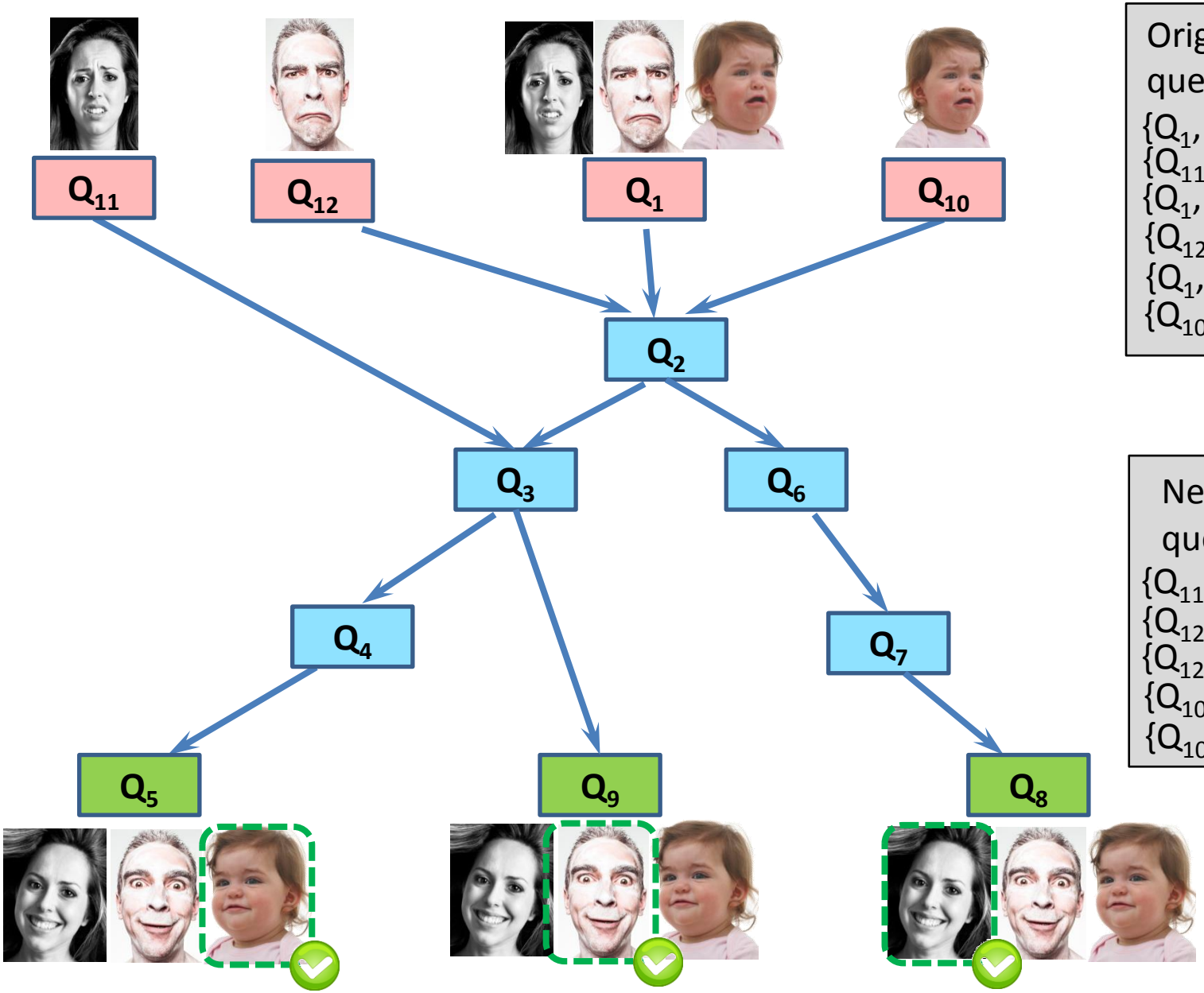
П-И-С-А-Т-Е-Л-Ь

П-И-С-А-Л



TB-Structure (merged Prefix & Suffix forests):

"collective or individual confusion" – "COLLABORATIVE satisfaction"



Original (Collected) query trails:

$\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
$\{Q_{11}, Q_3, Q_9\}$
$\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
$\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
$\{Q_1, Q_2, Q_3, Q_9\}$
$\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

New (**inferred**) query trails:

$\{Q_{11}, Q_3, Q_4, Q_5\}$
$\{Q_{12}, Q_2, Q_3, Q_9\}$
$\{Q_{12}, Q_2, Q_6, Q_7, Q_8\}$
$\{Q_{10}, Q_2, Q_3, Q_4, Q_5\}$
$\{Q_{10}, Q_2, Q_3, Q_9\}$

Batch preprocessing of trails before feeding TB-structure (because order matters):

using "prefix-suffix similarity" function

$$SIMPS_{T_x,T_y} = \frac{|T_x \cap^P T_y| + |T_x \cap^S T_y|}{|T_x| + |T_y|}$$

$|T_x \cap^P T_y|$ - longest common prefix length

$|T_x \cap^S T_y|$ - longest common suffix length

T_x: { Q_1, Q_2, Q_3, Q_4, Q_5, Q_6 }

T_y: { Q_1, Q_2, Q_3, Q_7, Q_8, Q_5, Q_6 }

$$SIMPS_{T_x,T_y} = \frac{5}{13} \approx 0.3846$$

Original (Collected) query trails:

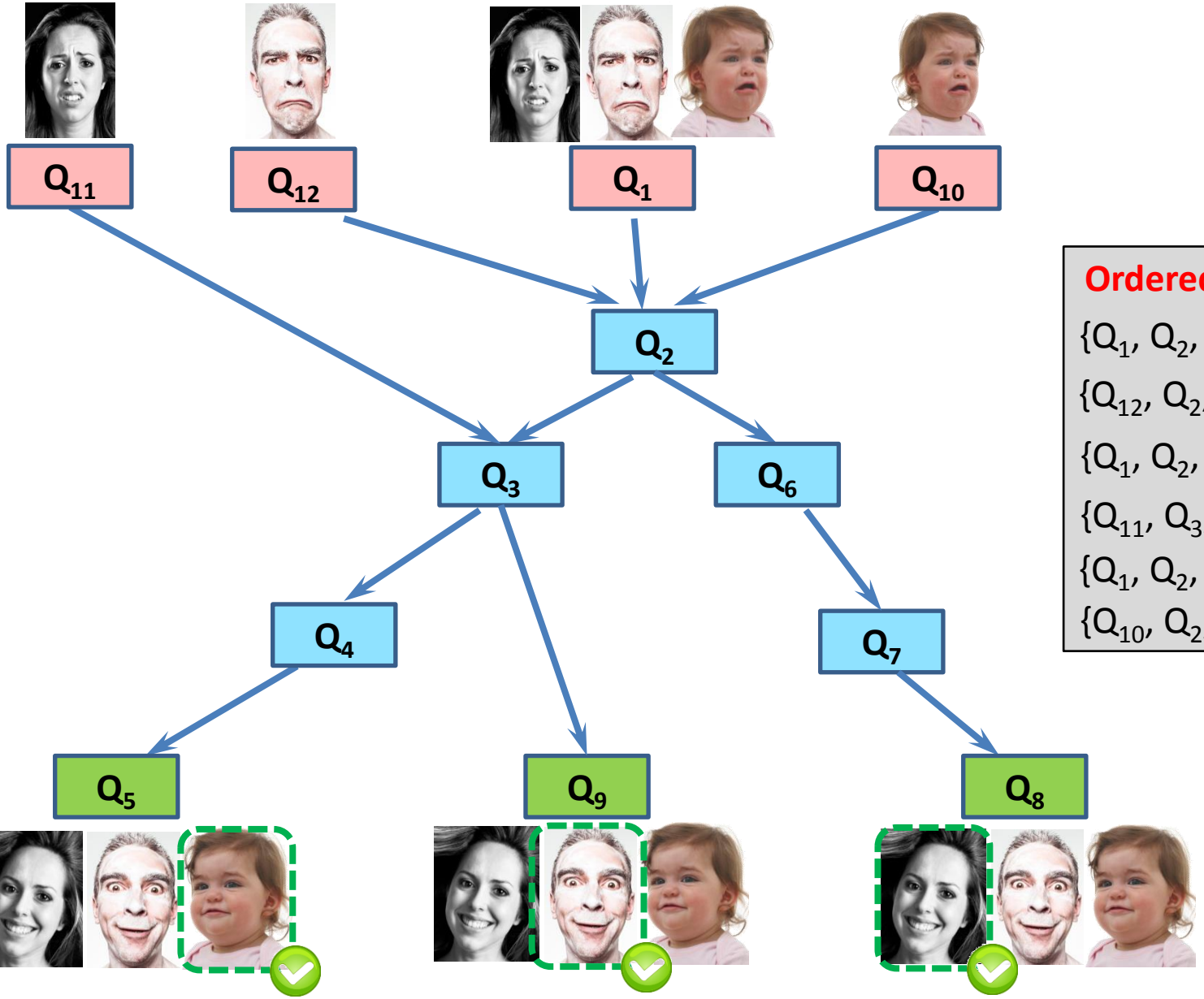
{Q_1, Q_2, Q_3, Q_4, Q_5}
{Q_{11}, Q_3, Q_9}
{Q_1, Q_2, Q_6 , Q_7 , Q_8}
{Q_{12}, Q_2, Q_3, Q_4, Q_5}
{Q_1, Q_2, Q_3, Q_9}
{Q_{10}, Q_2, Q_6, Q_7, Q_8}

Ordered query trails:

{Q_1, Q_2, Q_3, Q_4, Q_5}
{Q_{12}, Q_2, Q_3, Q_4, Q_5}
{Q_1, Q_2, Q_3, Q_9}
{Q_{11}, Q_3, Q_9}
{Q_1, Q_2, Q_6 , Q_7 , Q_8}
{Q_{10}, Q_2, Q_6, Q_7, Q_8}

How we got this TB-Structure:

Step-by-Step structure feeding



Ordered query trails:

$\{Q_1, Q_2, Q_3, Q_4, Q_5\}$

$\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$

$\{Q_1, Q_2, Q_3, Q_9\}$

$\{Q_{11}, Q_3, Q_9\}$

$\{Q_1, Q_2, Q_6, Q_7, Q_8\}$

$\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

How we got this TB-Structure:

Step 1



Q_1

Q_2

Q_3

Q_4

Q_5



Ordered query trails:

- $\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_1, Q_2, Q_3, Q_9\}$
- $\{Q_{11}, Q_3, Q_9\}$
- $\{Q_1, Q_2, Q_6 , Q_7 , Q_8\}$
- $\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

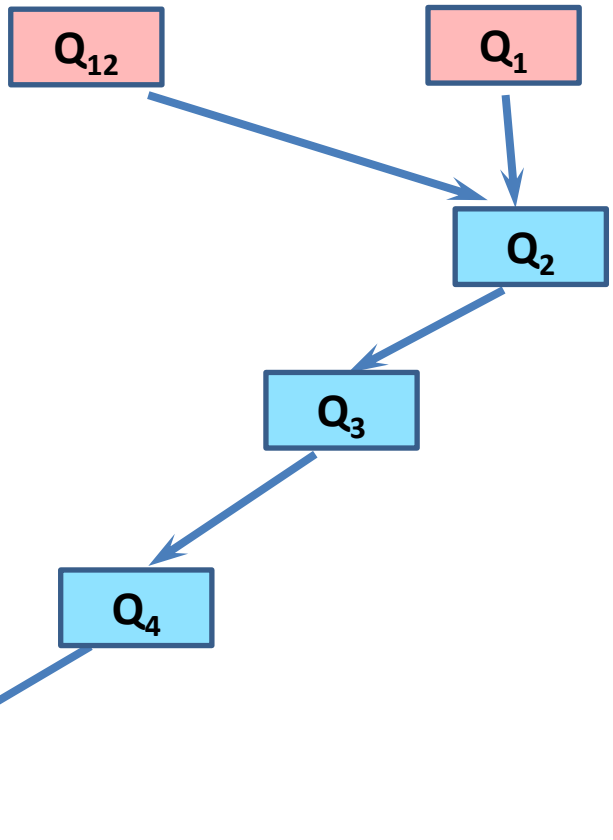
How we got this TB-Structure:

Step 2









Ordered query trails:

- $\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_1, Q_2, Q_3, Q_9\}$
- $\{Q_{11}, Q_3, Q_9\}$
- $\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
- $\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

How we got this TB-Structure:

Step 3





Q_{12} Q_1 Q_2 Q_3 Q_4 Q_5 Q_9





Ordered query trails:

- $\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_1, Q_2, Q_3, Q_9\}$
- $\{Q_{11}, Q_3, Q_9\}$
- $\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
- $\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

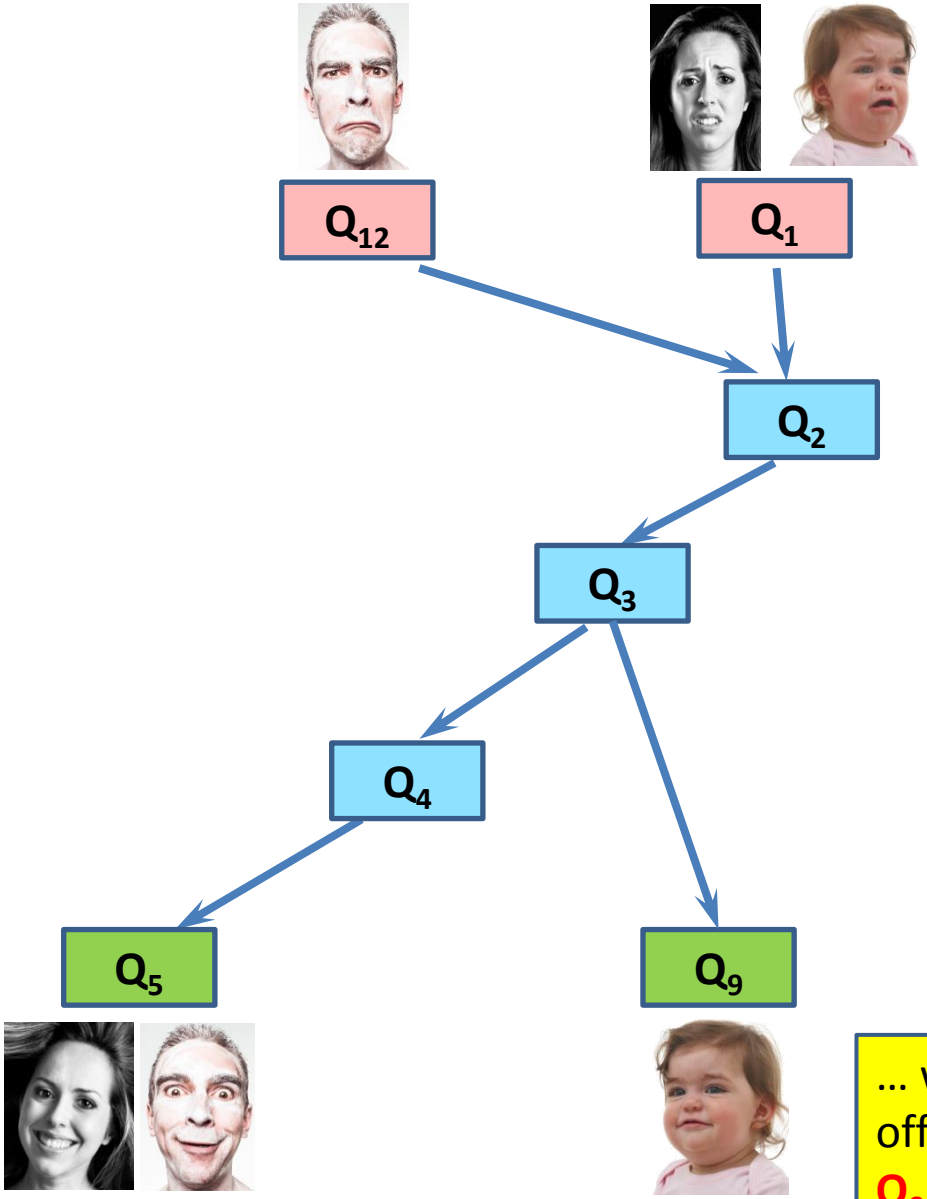
NOTICE NEW (INFERRED) TRAIL:

$\{Q_{12}, Q_2, Q_3, Q_9\}$

… which means that for the entry Q_{12} we may offer also search outcomes of the implicit query Q_9 (as well as, of course, of the explicit one Q_5)

How we got this TB-Structure:

Step 3* ("Collaborative Filtering" effect?)



NOTICE EFFECT aka "Collaborative Filtering" !!!

The underlying assumption of the collaborative filtering approach is that if a person *A* has the same "satisfaction" as a person *B* on an issue *X* (i.e., on the content returned by a search engine), then *A* is more likely to be satisfied on a different issue *Y*, which has already satisfied *B*, than to have the same satisfaction on *Y* as a person chosen randomly.

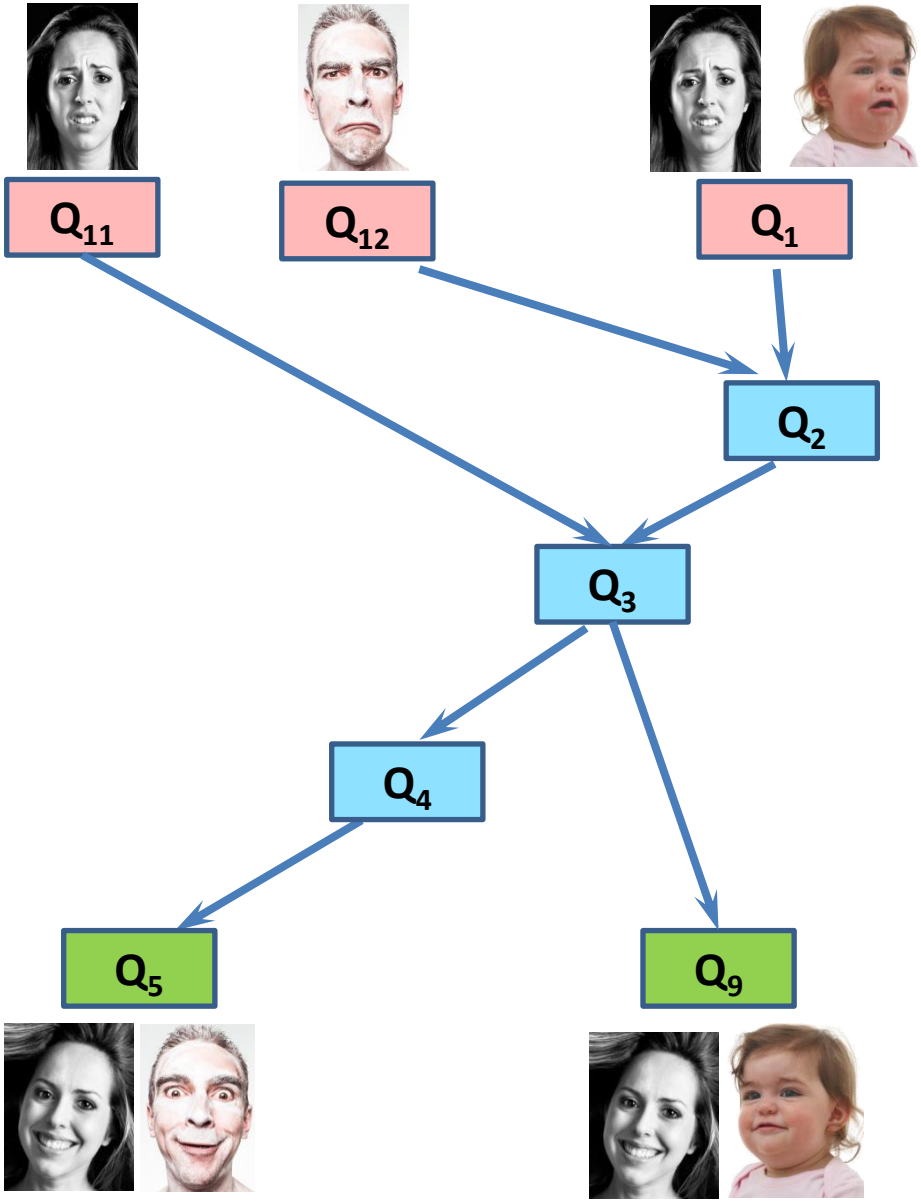
NOTICE NEW (INFERRED) TRAIL:

$\{Q_{12}, Q_2, Q_3, Q_9\}$

… which means that for the entry **Q_{12}** we may offer also search outcomes of the implicit query **Q_9** (as well as, of course, of the explicit one **Q_5**)

How we got this TB-Structure:

Step 4



Ordered query trails:

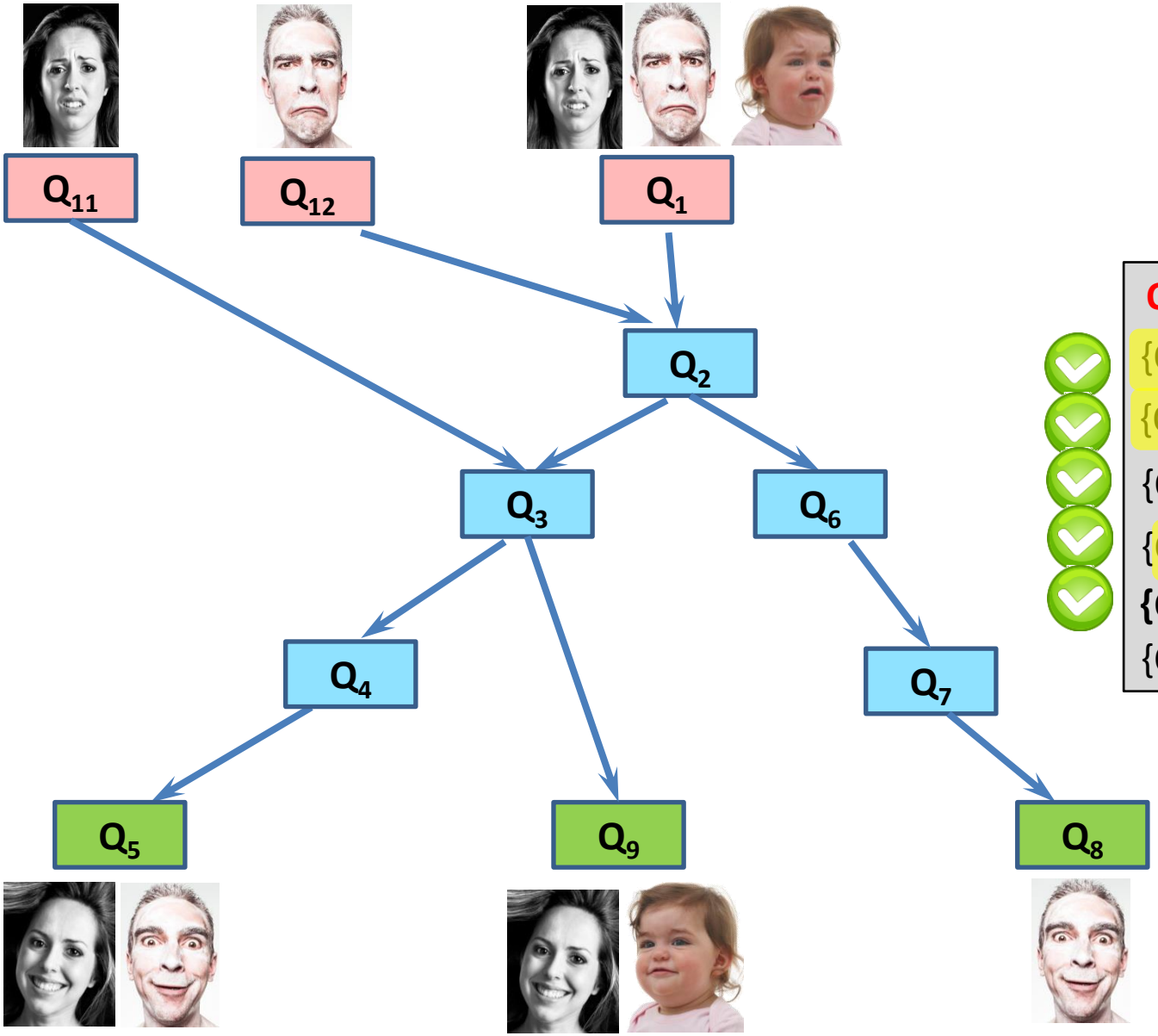
- $\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_1, Q_2, Q_3, Q_9\}$
- $\{Q_{11}, Q_3, Q_9\}$
- $\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
- $\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

NOTICE NEW (INFERRED) TRAIL:

$\{Q_{11}, Q_3, Q_4, Q_5\}$

How we got this TB-Structure:

Step 5



Ordered query trails:

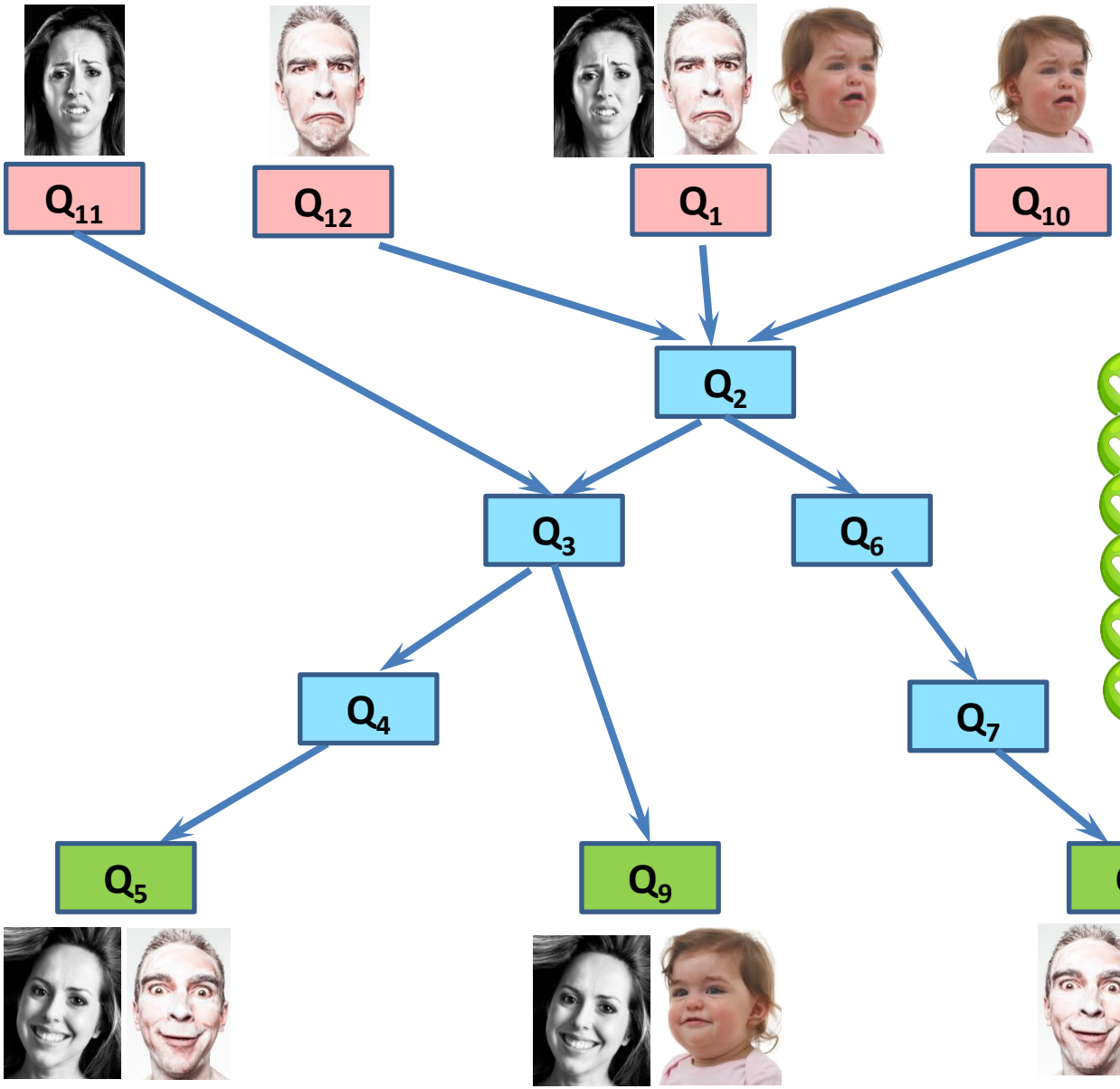
- $\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_1, Q_2, Q_3, Q_9\}$
- $\{Q_{11}, Q_3, Q_9\}$
- $\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
- $\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

NOTICE NEW (INFERRED) TRAIL:

$\{Q_{12}, Q_2, Q_6, Q_7, Q_8\}$

How we got this TB-Structure:

Step 6



Ordered query trails:

- $\{Q_1, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$
- $\{Q_1, Q_2, Q_3, Q_9\}$
- $\{Q_{11}, Q_3, Q_9\}$
- $\{Q_1, Q_2, Q_6, Q_7, Q_8\}$
- $\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

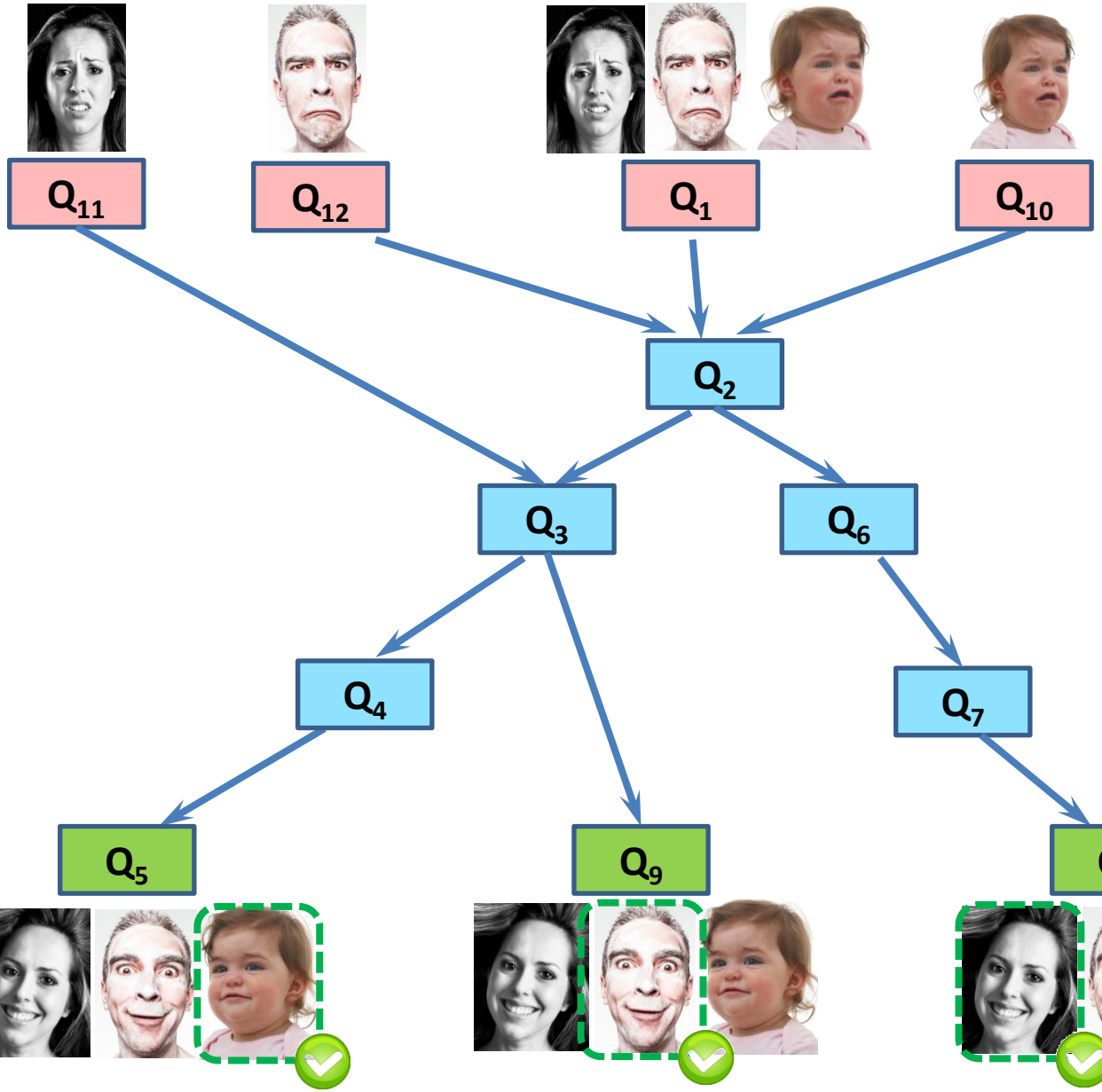
NOTICE 2 NEW (INFERRED) TRAILS:

$\{Q_{10}, Q_2, Q_3, Q_4, Q_5\}$

$\{Q_{10}, Q_2, Q_3, Q_9\}$

How we got this TB-Structure:

(finally, notice collaborative satisfaction nodes due to inferred trails)



Ordered query trails:

$\{Q_1, Q_2, Q_3, Q_4, Q_5\}$

$\{Q_{12}, Q_2, Q_3, Q_4, Q_5\}$

$\{Q_1, Q_2, Q_3, Q_9\}$

$\{Q_{11}, Q_3, Q_9\}$

$\{Q_1, Q_2, Q_6, Q_7, Q_8\}$

$\{Q_{10}, Q_2, Q_6, Q_7, Q_8\}$

5 INFERRED TRAILS:

$\{Q_{12}, Q_2, Q_3, Q_9\}$

$\{Q_{11}, Q_3, Q_4, Q_5\}$

$\{Q_{12}, Q_2, Q_6, Q_7, Q_8\}$

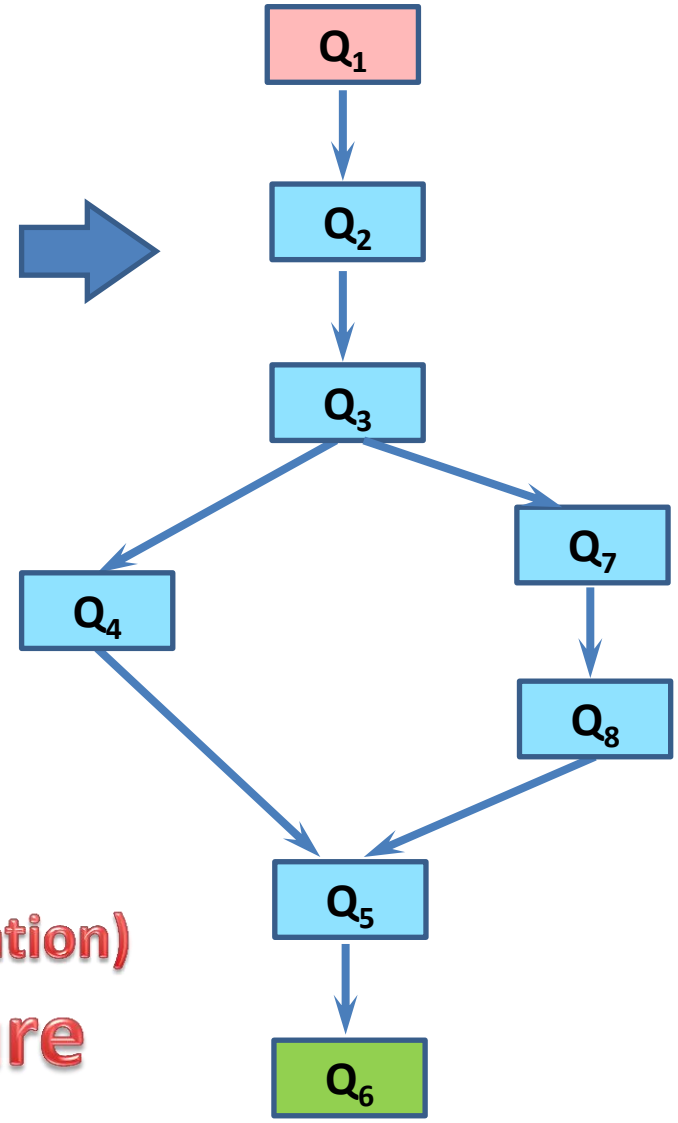
$\{Q_{10}, Q_2, Q_3, Q_4, Q_5\}$

$\{Q_{10}, Q_2, Q_3, Q_9\}$

Trails merged simultaneously from prefix and suffix: of course possible, no problem!

{ Q_1, Q_2, Q_3, Q_4, Q_5, Q_6 }

{ Q_1, Q_2, Q_3, Q_7, Q_8, Q_5, Q_6 }

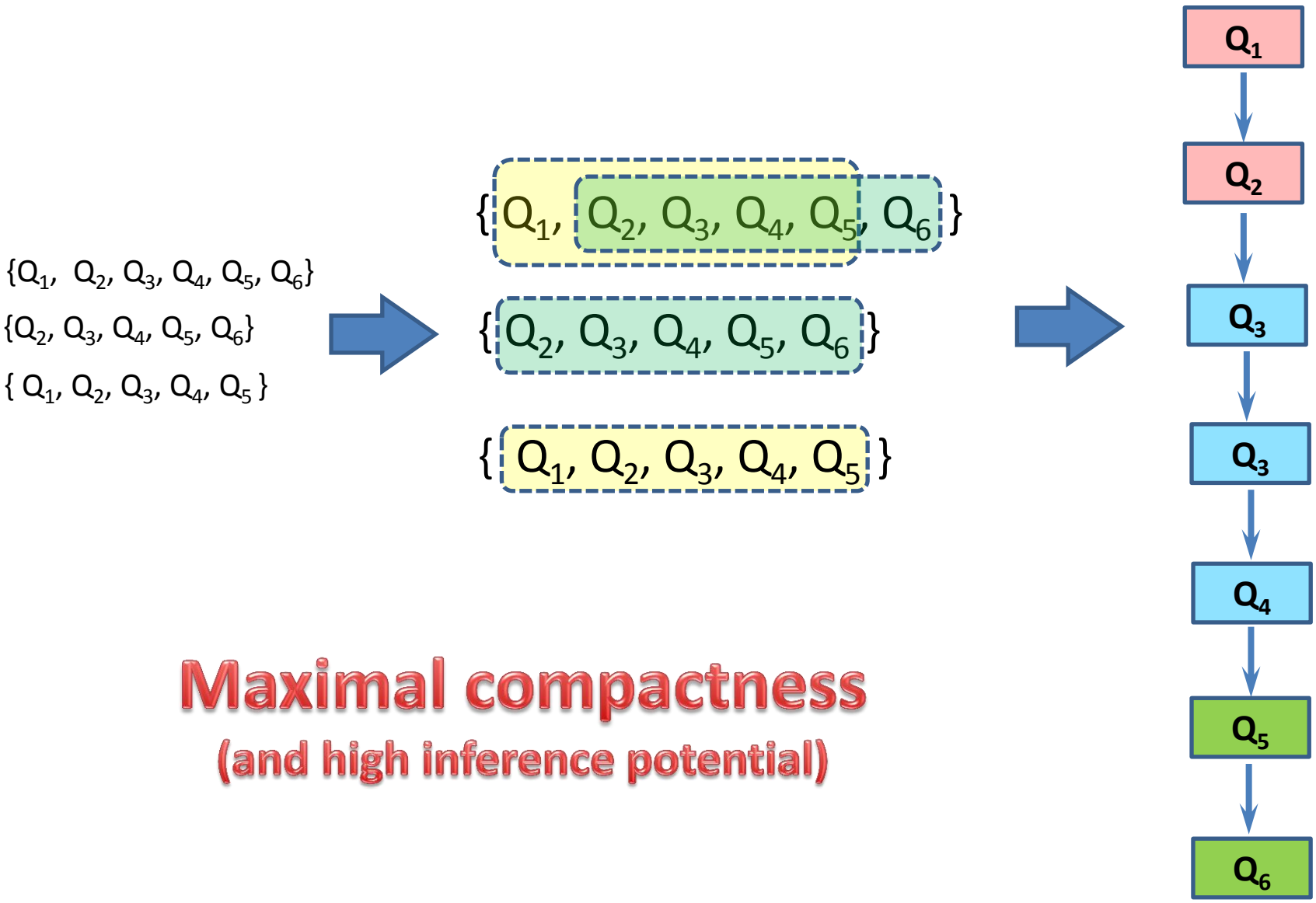


Unordered loops

(alternative paths from origin to destination)

are possible in TB-Structure

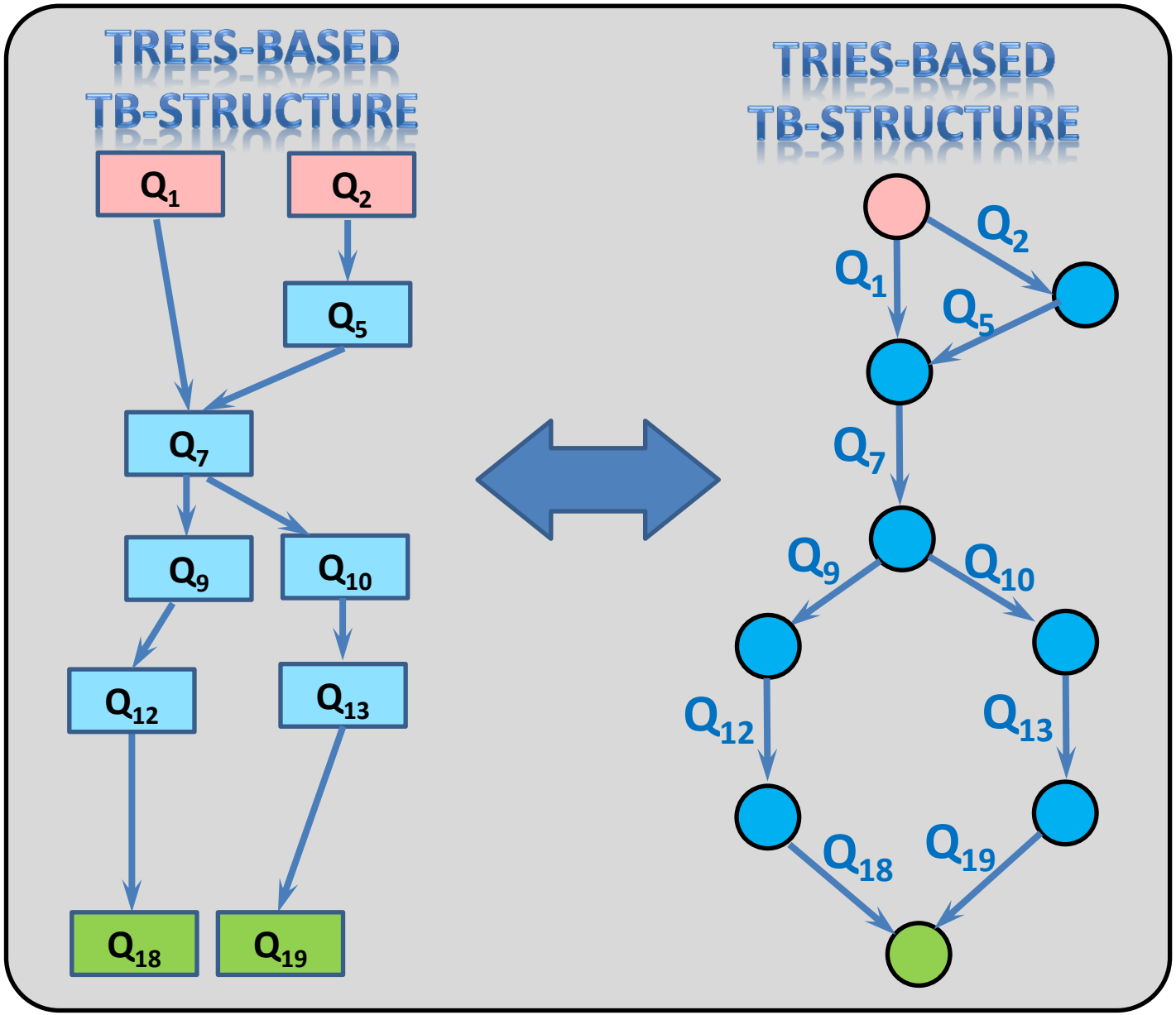
Nested prefixes or/and suffixes: of course possible, no problem!



Maximal compactness

(and high inference potential)

Trails: Trees vs. Tries



Experiments with TB-Structure (Data Generation)

To test the properties of a TB-structures the first round of experiments has been run on an ***automatically generated artificial data*** set.

The *automatic generation of trails* was performed in two different ways:

1. With a ***uniform generator***, which generated trails by choosing query symbols Q_i from the so called "alphabet" {**Q**} uniformly at random and constructing trails of the length chosen randomly between l_{min} and l_{max} ;
2. With a ***random traversal generator*** which created a graph of the query symbols by placing each symbol in a node and adding a directed edge from each node to a fixed number of randomly chosen other nodes. A trail was generated by the graph traversal from a randomly chosen starting node.







Experiments with TB-Structure (Data Settings)

We automatically generated 5391 restricted* TB-structures and 5391 fail-over** structures with different settings.

We varied the minimal trail length $l_{\min}$ from 5 to 55 nodes, the maximal length $l_{\max}$ varies from 5 to 60 nodes, the step of the length for each new set of generated structures was 5 nodes;

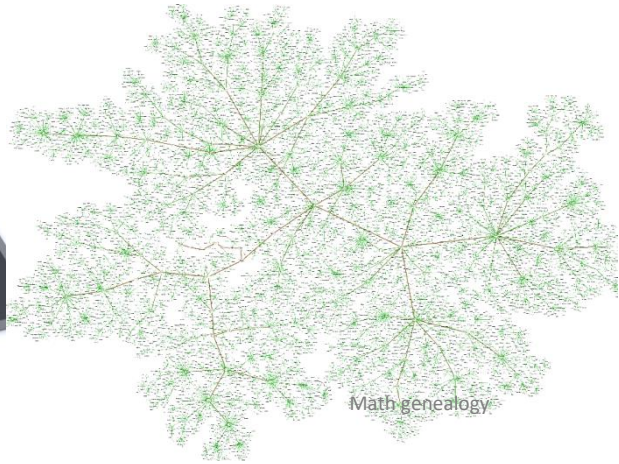
A number of symbols in the query alphabet varies from 20 minimum to 1280 maximum;

A number of initial trails varies from 100 to 51200.

* Trails in restricted structures does not contain the same symbol (query) several times in various places (natural assumption);
** Trails in fail-over structures does not have any limitations (may lead to so called “toxic” cases).







Experiments with TB-Structure

(verifying new-trails' "generative power" (inference capacity), which also means storage "compactness")

An important property of the TB structure verified by experiments is its generative power – ability to infer new trails, implicit in the initial collection of trails. Such property shows both the inference power and the compactness of the structure.

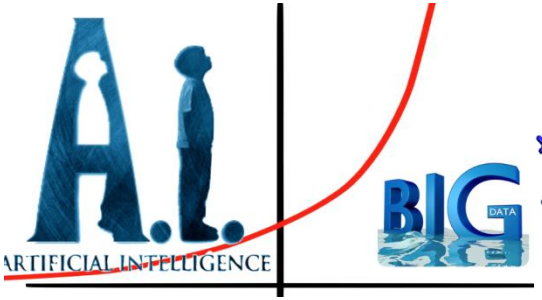
The experiments shown a non-linear, exponential-like dependency between the number of initial trails and newly generated ones.

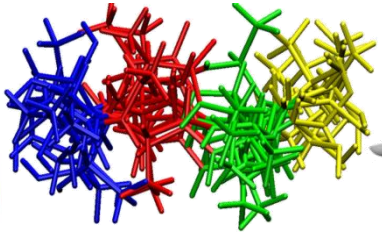
The maximum number of generated trails was achieved in case of the bigger difference between the minimal possible length of trails and their maximal possible length (l_{min} = 5 ; l_{max} = 60) and the smallest possible alphabet (= 20). The biggest explosion of new trails was observed in a fail-over structure. In many cases the number of generated trails was so large that we were unable to count it at in reasonable time*.

* The combinatorial explosion is not critical in our case because usually real search tasks imply shorter query trails than what we used in experiments, and a large alpha-bet. TB-structures constructed with these settings are more compact and suffer less from explosion of newly generated trails.



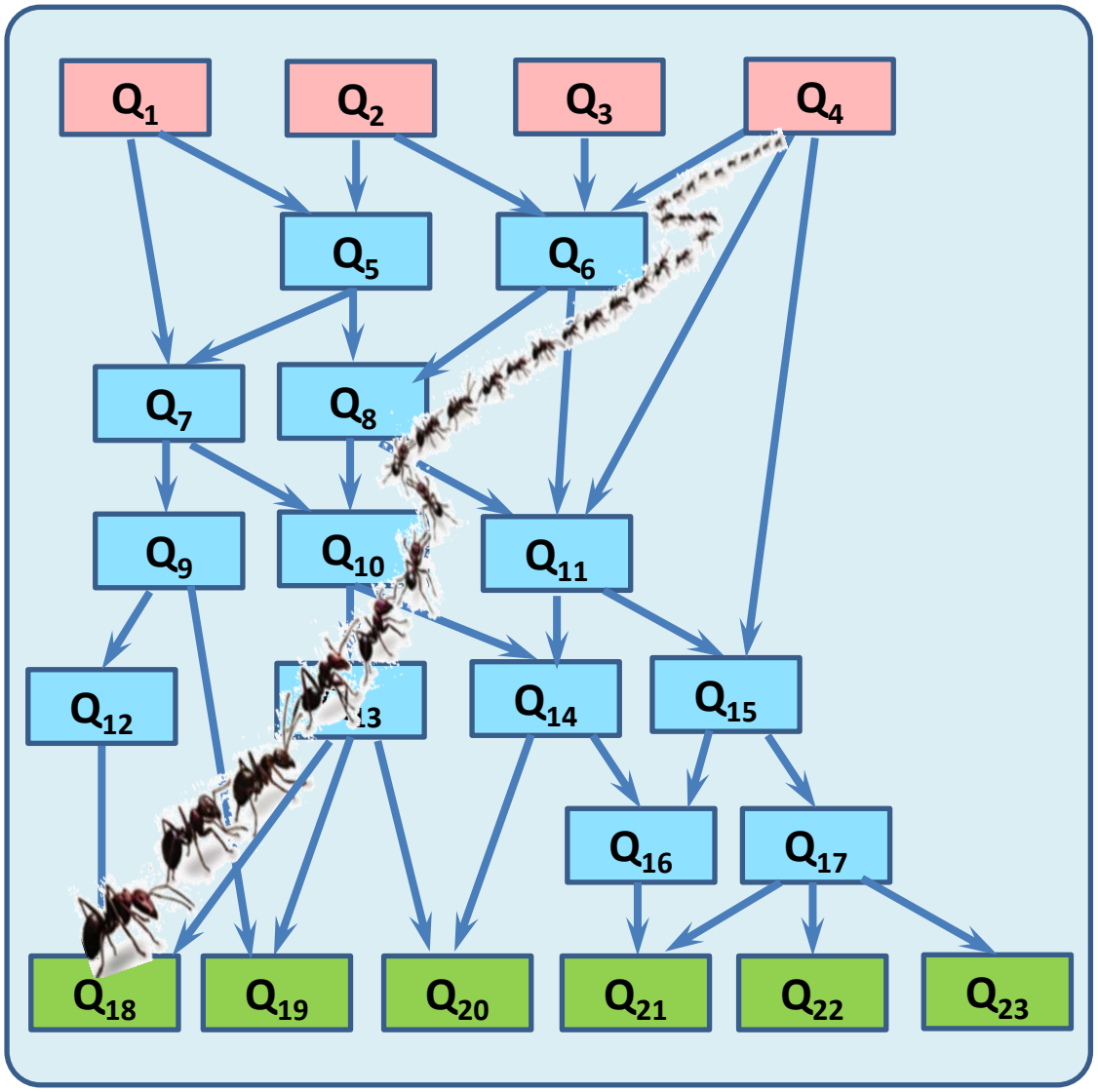


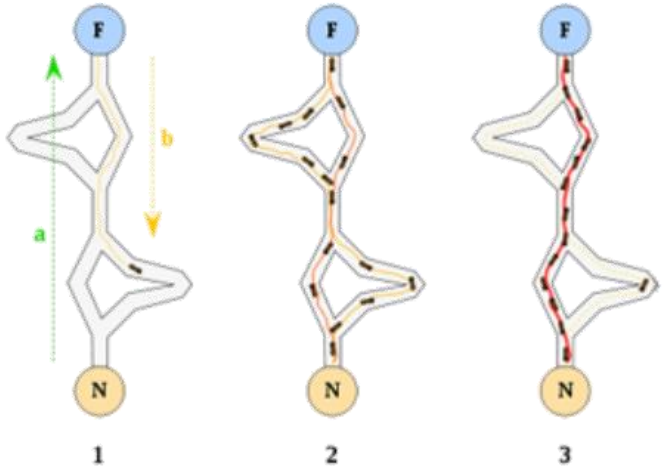






Next Stage: SWARM-TB-Query-Structure

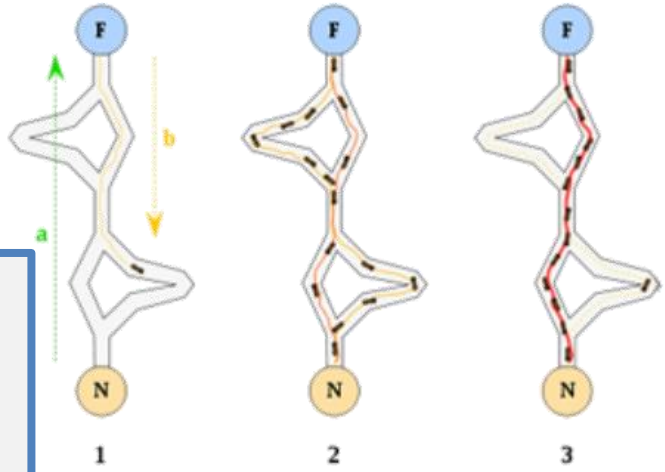






SWARM-TB-Query-Structure

HOW WE MAY USE PHEROMONE?



Ant Colony Optimization scheme:

[D. Merkle & M. Middendorf]



Initialize pheromone values
repeat
  for ant $k \in \{1, \ldots, m\}$
    construct a solution
  endfor
  forall pheromone values **do**
    decrease the value by a certain percentage {**evaporation**}
  endfor
  forall pheromone values corresponding to good solutions **do**
    increase the value {**intensification**}
  endfor
until stopping criterion is met



SWARM-TB-Query-Structure

Traveling Salesperson Problem ACO scheme:

[D. Merkle & M. Middendorf]



Initialize pheromone values
repeat
  for ant $k \in \{1, \ldots, m\}$ {solution construction}
    $S := \{1, \ldots, n\}$ {set of selectable edges}
    choose edge i with probability p_{0i}
    repeat
      choose edge $j \in S$ with probability p_{ij}
      $S := S - \{j\}$
      $i := j$
    until $S = \emptyset$
  Endfor
  forall i, j **do**
    $\tau_{ij} := (1 - \rho) \cdot \tau_{ij}$ {evaporation}
  endfor
  forall i, j in iteration best solution **do**
    $\tau_{i,j} := \tau_{ij} + \Delta$ {intensification}
  endfor
until stopping criterion is met

$$p_{ij} := \frac{\tau_{ij}}{\sum_{z \in S} \tau_{iz}} \quad \forall j \in S$$



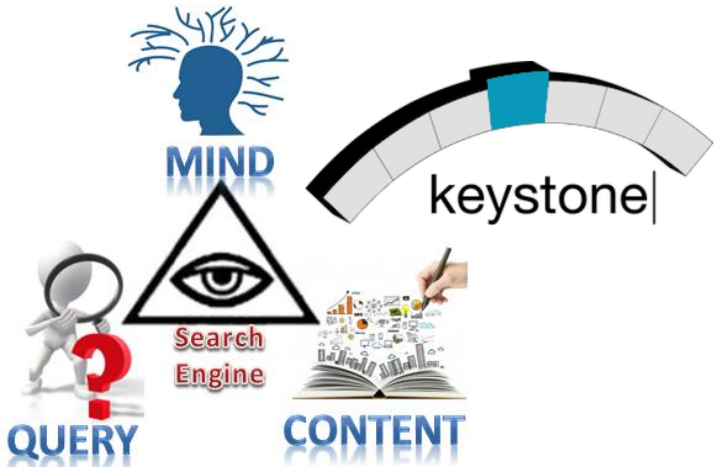
ACO variables and parameters.

τ_{ij}	Pheromone value
m	Number of ants per iteration
ρ	Evaporation rate
Δ	Amount of pheromone added during pheromone intensification

Summary and Conclusions

- Exploratory search optimization benefits from capturing and processing collective search experience;
- Collective behavior contains hidden patterns that can be uncovered and used;
- TB-Structure contributes to the effective storage and operation over big data sequences, e.g., query trails of collective search experience;
- TB-structure is a smart data model used for compact storage of explicit query trails and inference of implicit trails useful for new users' intents prediction;
- TB-structure can be applied for various tasks implying sequential processes, configurations and plans in biology, medicine, industry, logistics, etc.;
- Experiments show that generative power of the proposed data structures is very high, in some cases we experience explosion of new implicit knowledge emergence;
- The structure will benefit from applying a swarm intelligence on top of it.







If no time for explicit questions, then explore:

http://www.cs.jyu.fi/ai/IKC-2016.pptx

for implicit answers …

