Data-Driven Crowdsourcing

Tova Milo

TEL AVIV UNIVERSITY

Data Everywhere

The amount and diversity of Data being generated and collected is exploding….

I will focus today on human knowledge

Think of humanity and its collective mind expanding…

Background - Crowd (Data) sourcing

The engagement of crowds of Web users for data procurement

Can we trust the crowd ?

We need to be careful …

- What questions to ask?
  [SIGMOD13, VLDB13, ICDT14, SIGMOD14 VLDB15, SIGMOD15, VLDB16,SIGMOD18]
- How to define & determine correctness of answers?
  [WWW12, EDBT15, ICDE17]
- Who to ask? how many people? how to best use the resources?
  [VLDB13, ICDT13, ICDE13, SIGMOD15 PODS'15 ,ICDT16, VLDB16,WEDBD18]

Data Mining

Probabilistic Data

Data Cleaning

Optimizations and Incremental Computation

Crowd Mining: Crowdsourcing in an open world

- Human knowledge forms an open world
- Goal: extract *interesting* and *important* patterns

  - **Health care:** Folk medicine, people's habits, doctor's intuition…
  - **Finance:** People's habits & preferences, consultant's intuition…
  - …

- What questions to ask?

Back to classic databases...

- Significant data patterns are identified using **data mining**
- A useful type of pattern: **association rules**

The classical supermarket example…

→

- Queries are dynamically constructed in the learning process
- **Is it possible to mine the crowd?**

Turning to the crowd

Let us model the history of every user as a *personal database*

Treated a sore throat with garlic and oregano leaves…
Treated a sore throat and low fever with garlic and ginger …
Treated a heartburn with water, baking soda and lemon…
Treated nausea with ginger, then experienced sleepiness…
…

- Every case = a *transaction* consisting of *items*
- Not recorded anywhere – a hidden DB
  - It is hard for people to recall many details about many transactions!
  - But … they can often provide summaries, in the form of personal rules

    "To treat a sore throat I often use garlic"

Two types of questions

- Free recollection (mostly simple, prominent patterns)

  → **Open questions**

  Tell me about an illness and how you treat it

  “I typically treat nausea with ginger infusion”

- Concrete questions (may be more complex)

  → **Closed questions**

  When you have both nausea and fever, how often do you use a ginger and honey infusion?

Use the two types interleavingly.

More Examples

Ann, a vacationer, is interested in finding child-friendly activities at an attraction in NYC, and a good restaurant nearby (plus relevant advice).

"You can play baseball in Central Park and eat at Maoz Vegetarian.
Tips: Apply for a ballfield permit online"

"You can go visit the Bronx Zoo and eat at Pine Restaurant.
Tips: Order antipasti at Pine.
Skip dessert and go for ice cream across the street"

A dietician may wish to study the culinary preferences in some population, focusing on food dishes that are rich in fiber

More Examples

Ann, a vacationer, is interested in finding child-friendly activities at an attraction in NYC, and a good restaurant nearby (plus relevant advice).

“You can play baseball in Central Park and eat at Maoz Vegetarian.

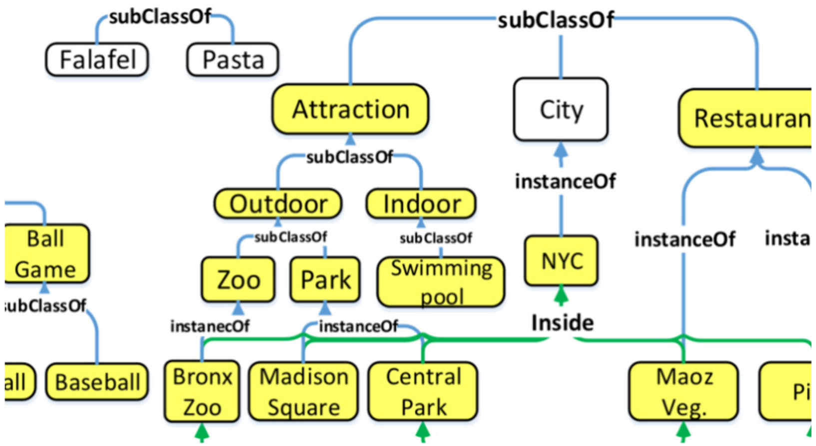
General knowledge:
- General truth, objective data, not associated with an individual
- *E.g., geographical locations*
- Can be found in a knowledge base or an ontology

When missing in the knowledge base, we can ask the crowd!

Individual knowledge:
- Related to the habits and opinions of an individual
- *E.g., travel recommendations*
- We can ask people about it

Crowd answers can be recoded in a knowledge base

Mixing General and Individual Knowledge



Given an ontology of general knowledge and a mining task

- Incrementally explore relevant patterns

```
{Ball_Game playAt Central_Park}
```

- Generate (closed and open) questions to the crowd about them

*How often do you **play ball games** at **Central Park**?*

*Which **ball games** do you **play** at **Central Park**? What else do you do at **Central Park**?*

- Evaluate the significance of the patterns and discover related ones

Pattern score = 0.6

```
{Baseball playAt Central_Park.
 Permit   getAt  "www.permits.org"}
```

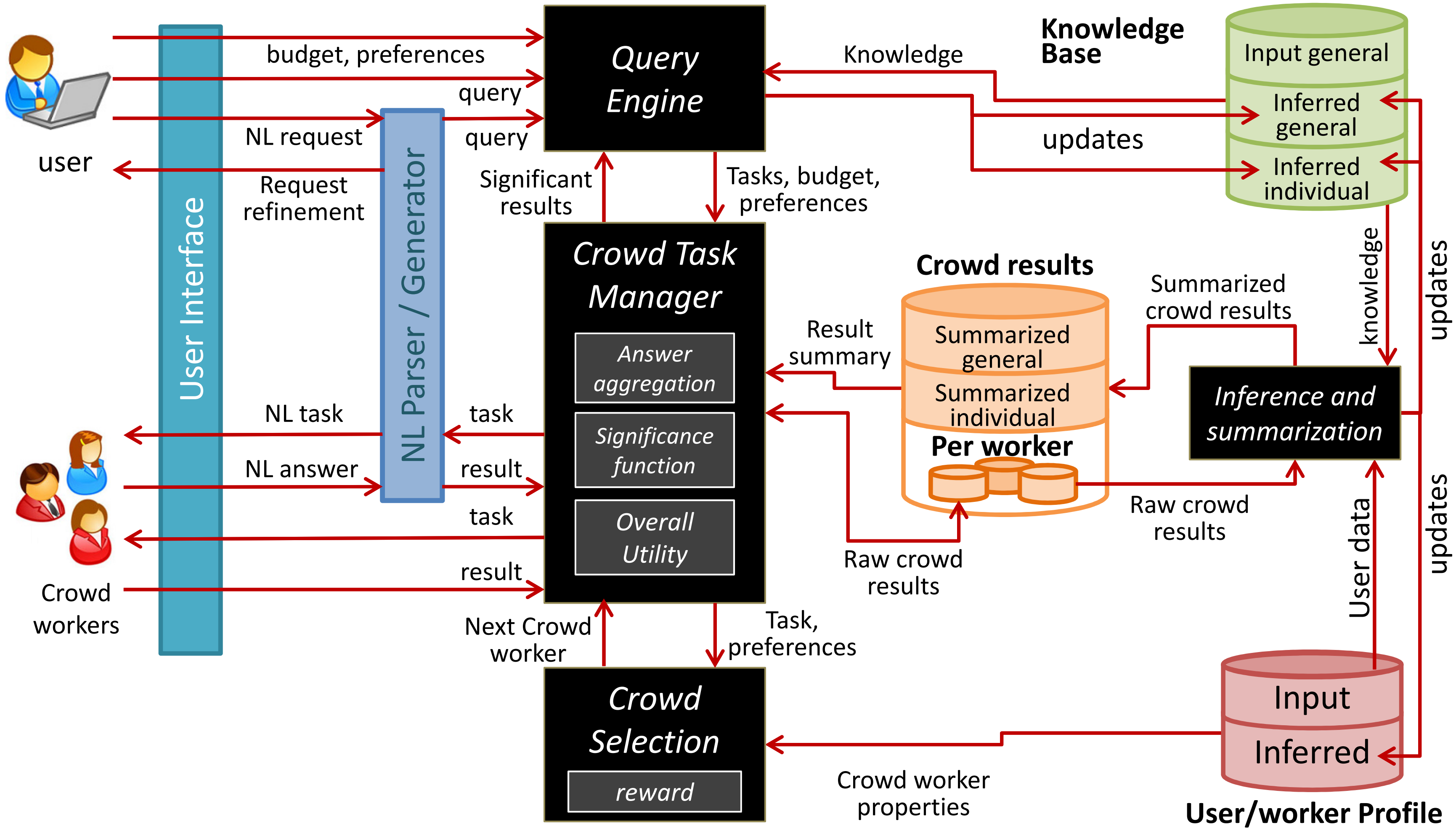
- Produce a concise output that summarizes the findings

Architecture Sketch

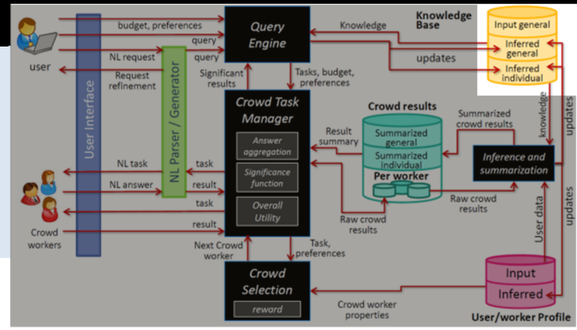
EU-FP7 ERC

MoDaS

Mob Data Sourcing



Knowledge Repository



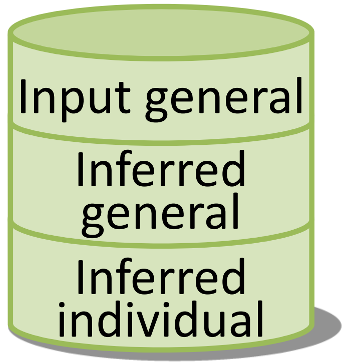
Different types of knowledge:

- A general knowledge base is input to the system

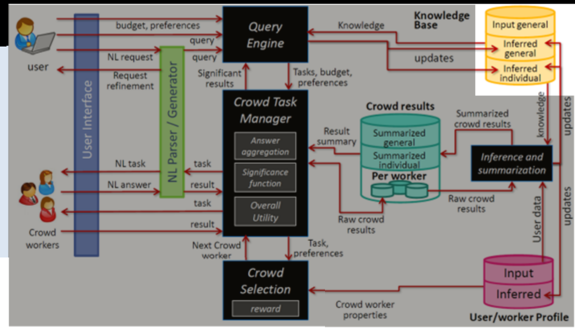
WordNet foursquare yago select knowledge

- Knowledge inferred in previous query evaluation

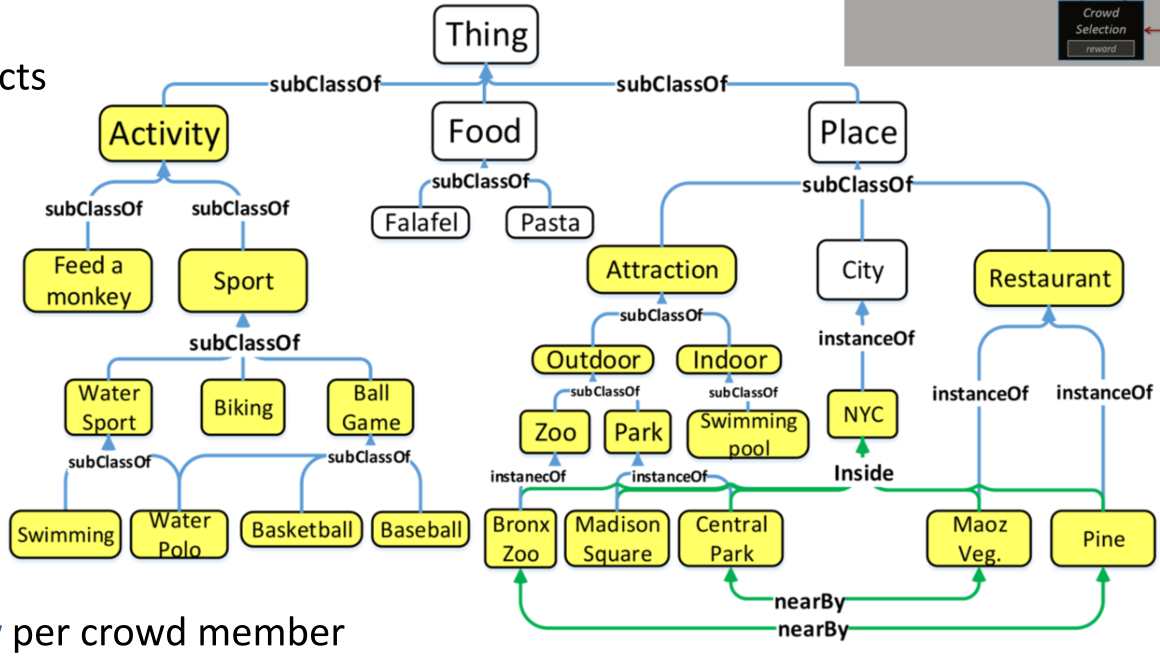
  - **General knowledge** – completes the knowledge base
    May be annotated with trust/error probability
  - **Individual knowledge** – more volatile
    may be annotated with user properties



Formal Model Based on RDF



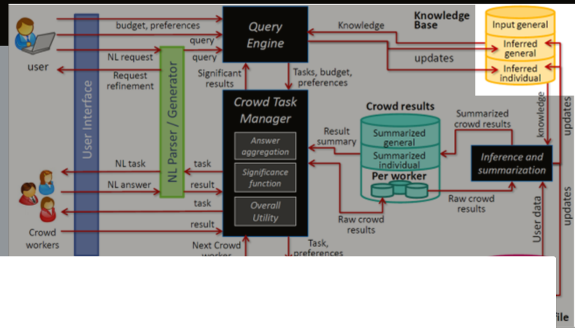
Ontology of general facts

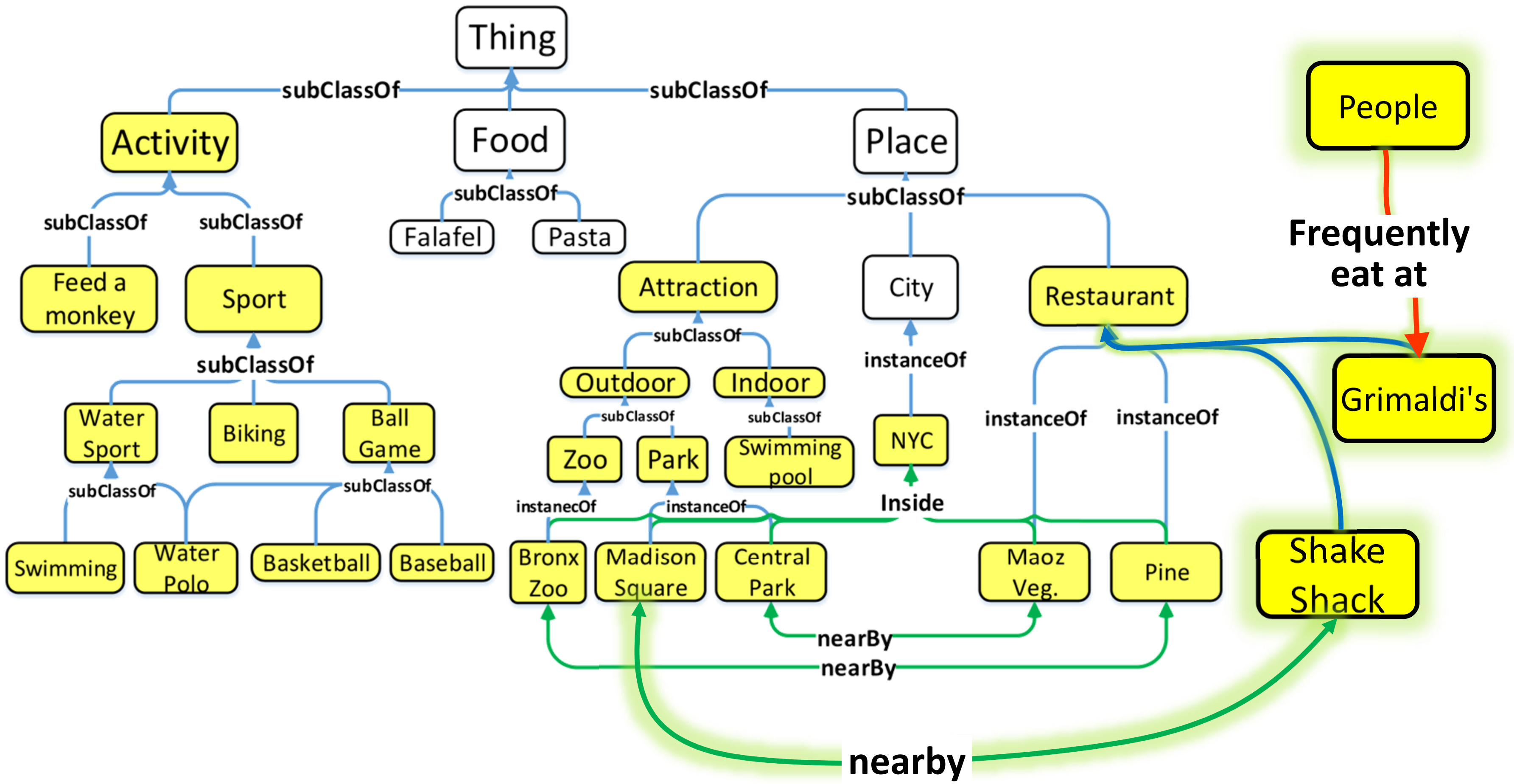


DB of personal history per crowd member

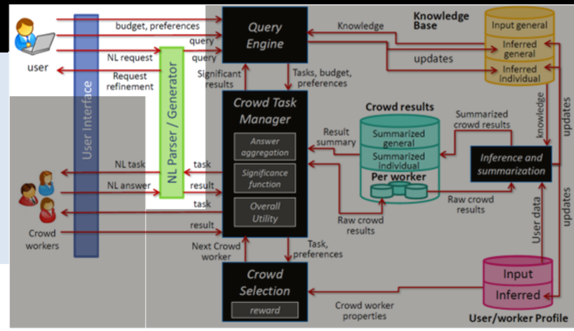
T1	I visited the Bronx Zoo and ate pasta at Pine on April 5th	`[Visit doAt Bronx_Zoo].` `[Pasta eatAt Pine]`
T2	I played basketball in Central Park on April 13th	`[Basketball playAt Central_Park]`
T3	I played baseball in Central Park and ate falafel at Maoz Veg. on April 27th	`[Baseball playAt Central_Park].` `[Falafel eatAt Maoz_Veg]`
	…	…

Formal Model Based on RDF





Enters a User…



- The user query may be formulated in a formal language

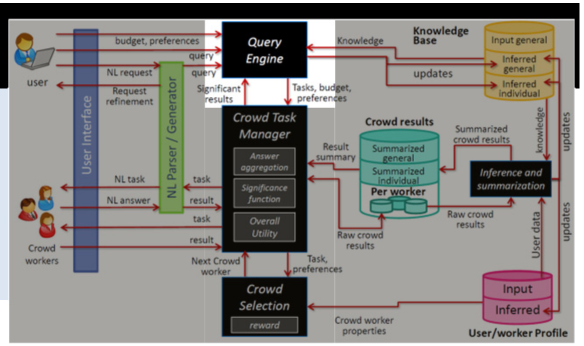
  E.g., OASSIS-QL is a SPARQL-based query language for crowd mining

  [SIGMOD'14, SIGMOD'15]

Find popular combinations of
an activity in a child-friendly attraction
at NYC and a restaurant nearby
(plus relevant advice)

```
SELECT VARIABLES
WHERE
    {$w subClassOf* Attraction
     $x instanceOf $w.
     $x inside     NYC.
     $y subClassOf* Activity.
     $z instanceOf Restaurant.
     $z nearBy     $x}
SATISFYING
     {$y+ doAt     $x.
      []  eatAt    $z.
     MORE}
     WITH SUPPORT = 0.03
```

Evaluation with the crowd



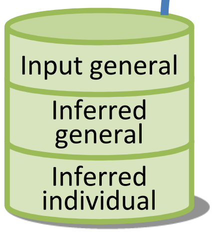

```
SELECT VARIABLES
WHERE
    {$w subClassOf* Attraction
     $x instanceOf $w.
     $x inside     NYC.
     $y subClassOf* Activity.
     $z instanceOf Restaurant.
     $z nearBy     $x}
SATISFYING
     {$y+ doAt     $x.
     []   eatAt    $z.
     MORE}
     WITH SUPPORT = 0.03
```

\$x= Central_Park,
\$y = Biking,
\$z = Maoz Veg.

Crowd task:
isSignificant({Biking doAt Central_Park},…)
Budget: \$0.5
User preferences: …

“How often do you go biking at Central Park and eat at Maoz Vegetarian?”

“Once a month.”
(support = 12/365)



What is a good algorithm?

How to measure the efficiency of Crowd Mining Algorithms ?

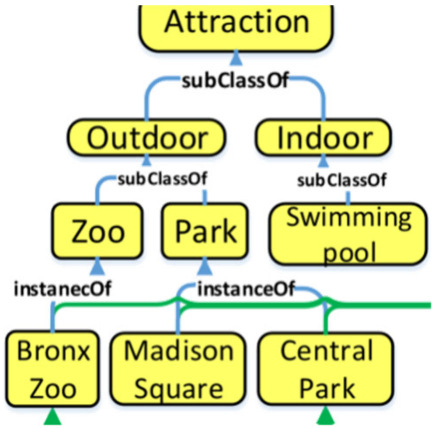
- Two distinguished cost factors:
  - **Crowd complexity:** # of crowd queries used by the algorithm
  - **Computational complexity:** the complexity of computing the crowd queries and processing the answers

[Crowd comp. lower bound is a trivial computational comp. lower bound]

- There exists a **tradeoff** between the complexity measures
  - Naïve questions selection -> more crowd questions

Efficient Query Evaluation Algorithm

- We want to minimize the number of questions to the crowd
- We define a semantic subsumption partial order over terms, facts, and fact-sets
- Used for
  - Pruning the search space
  - Compact output representation




```
Biking doAt Park
```

```
Biking doAt Central_Park
```

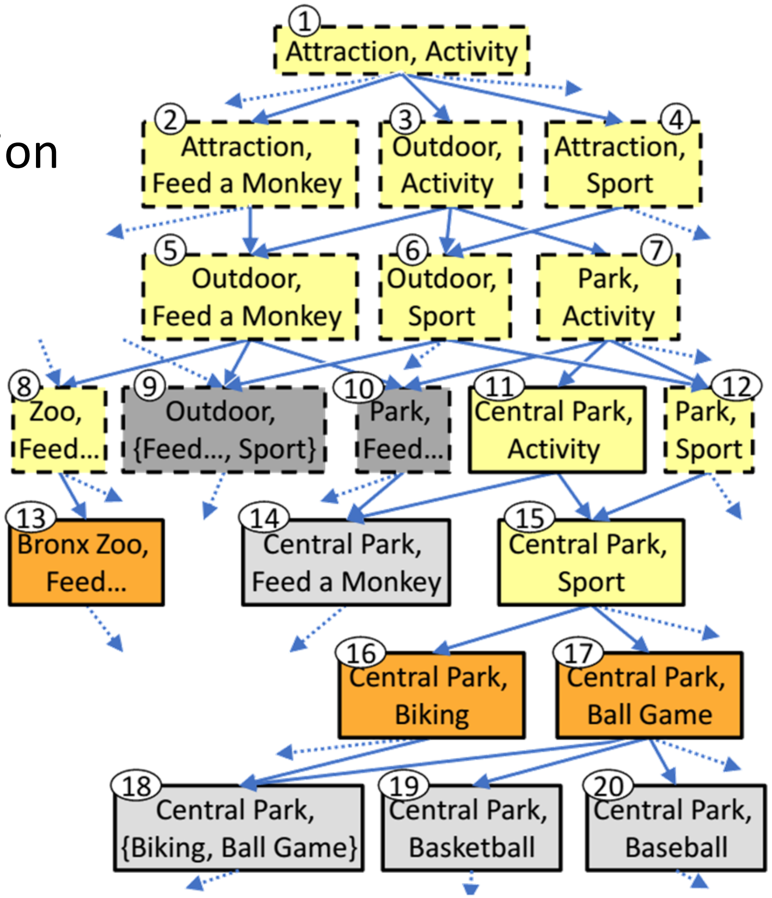
```
Biking     doAt   Central_Park.
Basketball playAt Central_Park
```

Efficient Query Evaluation Algorithm

The algorithm:

- Lazily construct the semantic subsumption partial order
- Traverse it in a top-down manner
- Prune insignificant parts

- See complexity analysis in the paper



Sometimes theory fails…

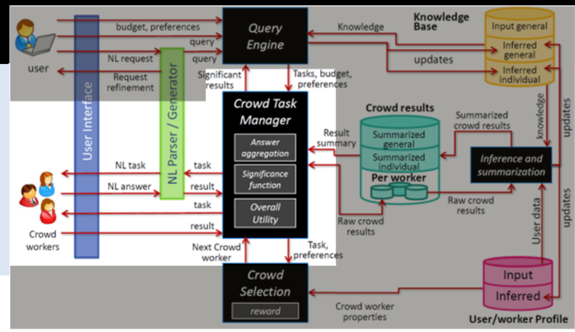
Practical (Crowd) Aspects of the Algorithm

- Asking a sequence of questions "in context"
- Quick pruning of irrelevant items by crowd members

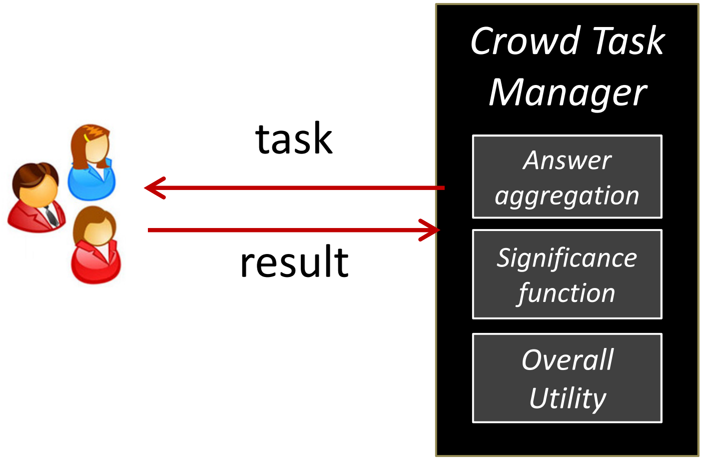
- Open questions – letting crowd members specify patterns

"What else do you do when you play basketball in Central Park?"

Crowd Task Manager



- Distributes tasks to crowd members
- Aggregates and analyzes the answers
- Dynamically decides what to ask next



Crowd task:
isSignificant({`Baseball doAt Central_Park`})
Budget: $0.5
User preferences: …

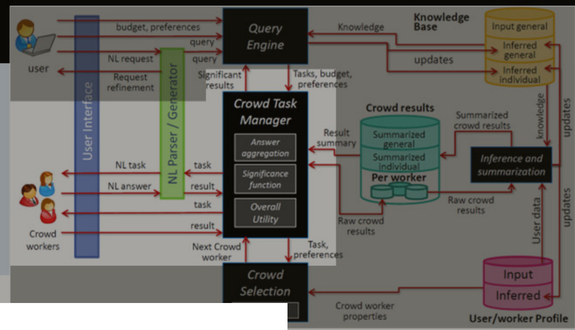
"How often do you play baseball at Central Park?"

Answer 1: never (score=0)

Answer 2: once a week (score=1/7)

Aggregation: estimated mean M
Significance: $\Pr(M \geq \Theta) \geq 0.5$
Overall utility: next question expected to reduce error probability by 0.1

Crowd Task Manager



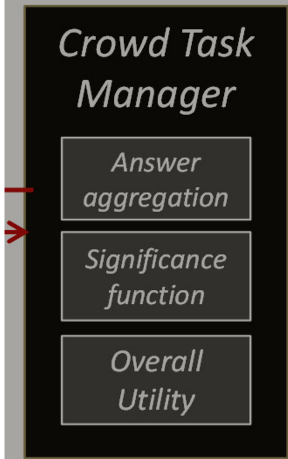
Aggregation, significance and utility choices depend on the type of data collected from the crowd.

For **individual** data, the aggregated answer should account for diverse opinions

- e.g., statistical modeling

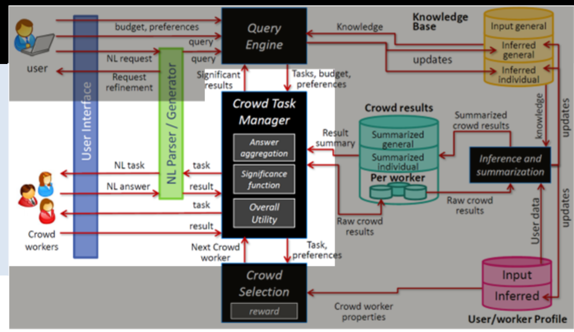
For **general** data the aggregated answer should reflect the truth

e.g., weighing by expertise, outlier filtering



mean M
≥ 0.5
tion expected
ty by 0.1

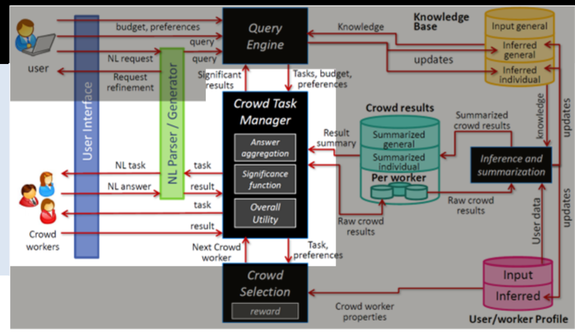
Crowd Task Manager



Theoretical research focuses on computing some DB operator with the crowd:
Sort, Top-K, Group-by, skyline, join…

- Some underlying ground truth is often assumed
- Typically Boolean questions (“is a>b ?”)
- Simple error model (user error < 0.5, given overall error bound)
- Mostly lower and upper bounds on the number of required questions

Sometimes theory fails (2)



Theoretical research focuses on computing some DB operator with the crowd:
Sort, Top-K, Group-by, skyline, join…

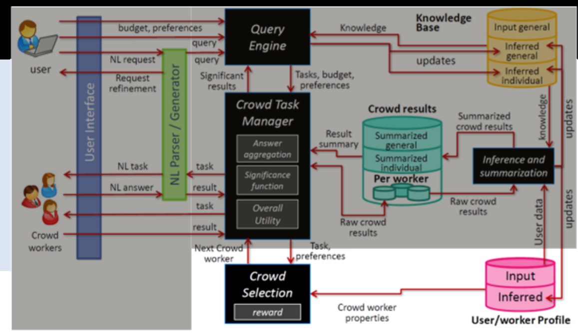
- Some underlying ground truth is often assumed
- Typically Boolean questions ("is a>b ?")
- Simple error model (user error < 0.5, given overall error bound)
- Mostly lower and upper bounds on the number of required questions

When high accuracy is required

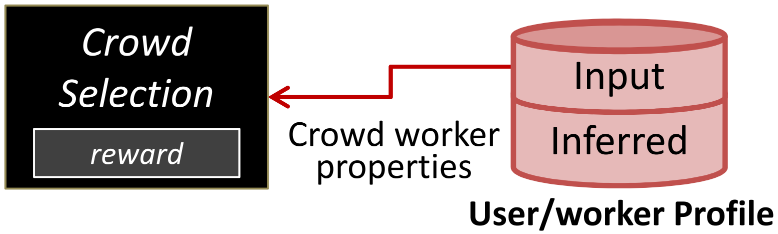
and the crowd error is high…

[Salable filtering algorithms. Groz, M. ICDT16]

Crowd Selection



- **By explicit user requirements:**
  crowd members should be NYC residents
- **By expertise:**
  based on past answers regarding general knowledge
- **By similarity to the user:** based on profiles,
  past questions and answers regarding individual data
  - Reward crowd members accordingly



Declarative User Selection

Ann plans to go to Australia in January, looks for a recommended Surfing beach near Sydney

```
SELECT VARIABLES $x
WHERE
    {$x instanceOf Place.
     $x near       Sydney}
SATISFYING
    {Surfing doAt $x.
     _       in January}
    ORDER BY DESC(SUPPORT)
    LIMIT 5
```

User Profiles

profile(Ann):		
SELF	livesIn	Paris
SELF	hasGender	Female
SELF	hasHobby	Photography
SELF	hasHobby	Bird_Watching
SELF	graduatedFrom	NYU

profile(Bill):		
SELF	livesIn	Sydney
SELF	hasGender	Male
SELF	hasHobby	Photography
SELF	graduatedFrom	University_of_Melbourne

profile(Carol):		
SELF	livesIn	Sydney
SELF	hasGender	Female
SELF	hasHobby	Art

Extended profile (User Answers)

answers(Ann):			
Fact-set	**Support**	**Frequency**	**IC**
Sport doAt Place	0.8	0.8	0.08
Surfing doAt Place	0.1	0.01	0.96
Surfing doAt Santa_Cruise	0.005	0.001	0.99
Volleyball doAt Place	0.001	0.02	0.93

answers(Bill):			
Fact-set	**Support**	**Frequency**	**IC**
Sport doAt Place	0.85	0.8	0.08
Surfing doAt Place	0.08	0.01	0.96
Surfing doAt Wanda_Beach. _ in January	0.012	0.0001	0.99
Volleyball doAt Place	0	0.02	0.93

answers(Carol):			
Fact-set	**Support**	**Frequency**	**IC**
Sport doAt Place	0.2	0.8	0.08
Surfing doAt Place	0.026	0.01	0.96
Surfing doAt Long_Reef. _ in January	0.013	0.00009	0.99
Drawing doAt Place	0.15	0.05	0.84

Declarative User Selection

```
ASSIGN BY Ann TO $u
FROM ontology WHERE
    {$x instanceOf Place.
     $x near       Sydney}
FROM profile($u) WHERE
    {SELF livesIn Sydney}
FROM tran($u) WHERE
    {{Surfing doAt []} WITH SUPPORT > 0.02
SIMILAR profile($u) TO profile(Ann)
    WITH SIMILARITY >= 0.75
SIMILAR tran($u) TO tran(Ann)
    WITH SIMILARITY >= 0.75
SIMILAR tran($u) TO
    {Surfing doAt $x.
     _       in January} AS surfhabit
    WITH SIMILARITY AS surfSim >= 0
ORDER BY surfSim LIMIT 1
```

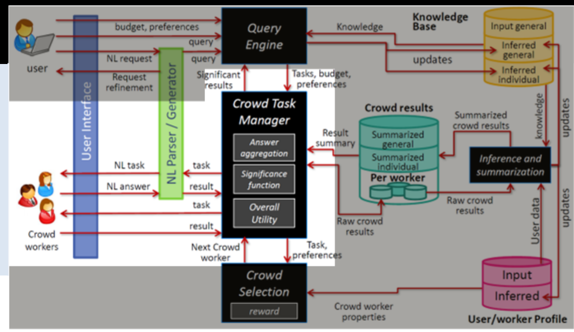
[places near Sydney]

Crowd members
who live in Sydney
and surf frequently

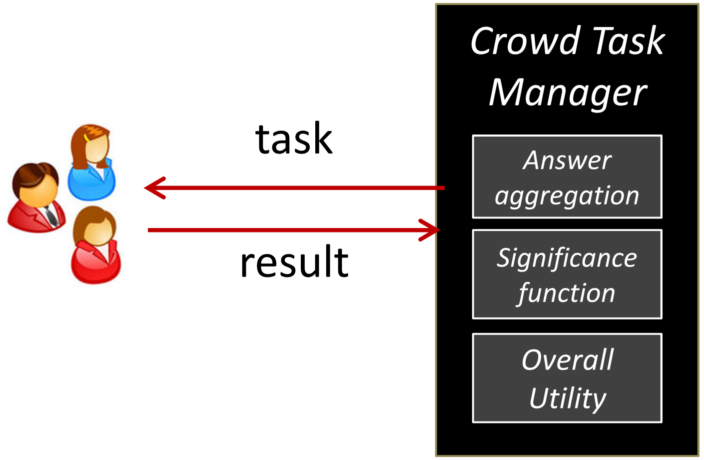
With profile and
past answers
similar to Ann's

Surfing near Sydney
In January (or similar)

Task Manager (revisited)



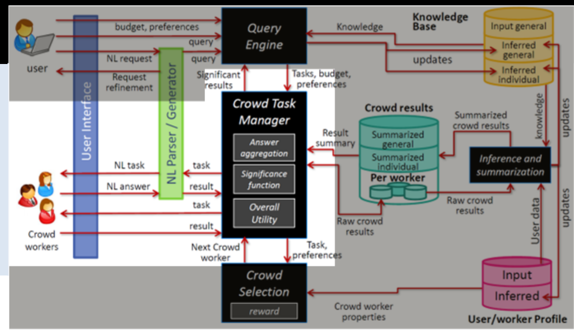
- Distributes tasks to selected crowd members
- Aggregates and analyzes the answers
- Dynamically decides what to ask next



Theoretical research here typical focuses on optimal task distribution

- Some crowd expertise levels are assumed/dynamically inferred
- Some tasks difficulty levels are assumed/dynamically inferred
- Some bound on the possible worker load is assumed
- Maximization of result quality while minimizing time/cost

Still, sometimes queries are too difficult...



Solution: Dismantle into easier ones, then Reassemble

Original query

new query 1 → Answer to query 1

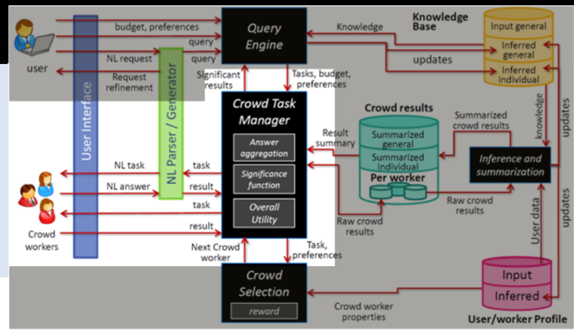
new query 2 → Answer to query 2

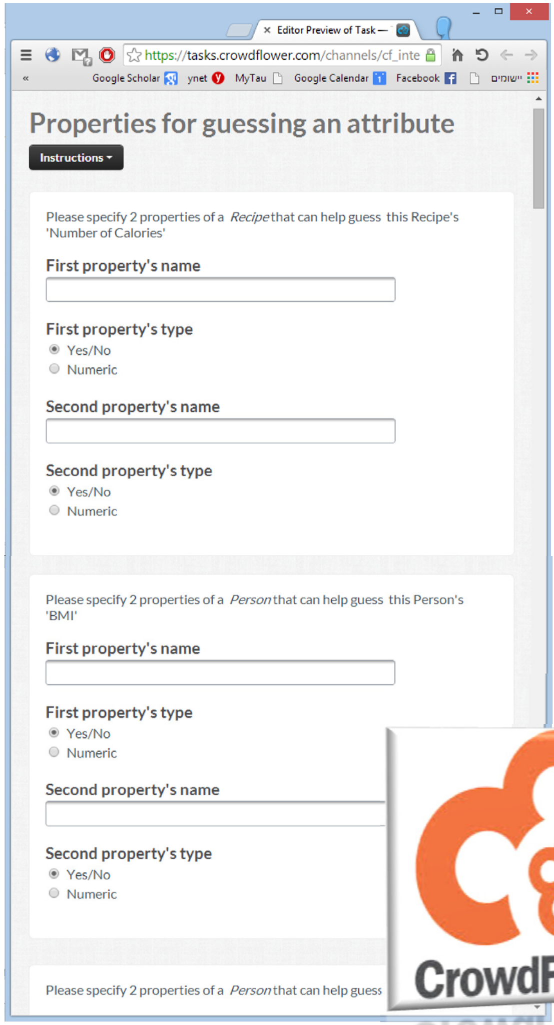
... → ...

Answer to original query

Can we do it all automatically with the crowd?

Examples





CrowdFlower

Person's age

- wrinkles, grey hair, old, height, good look, children, dark skin, has work, male, over 35, weight, glasses, …

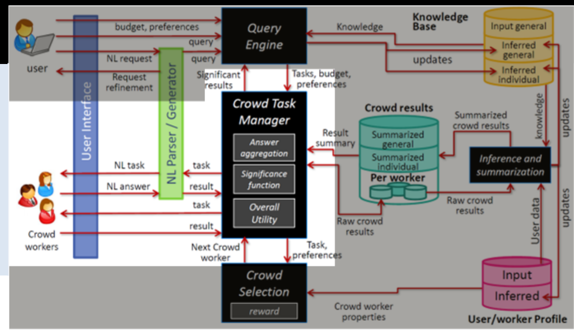
Recipe's #calories

- fat amount, #ingredients, healthy, portion size, sugar amount, vegetarian, oily, dietetic,…

House's price

- good location, age, size, #room, good neighborhood, good view, renovated, nice, good exterior condition, …

Dismantling queries



Input: Query ("Select BMI from Pictures") and Budget

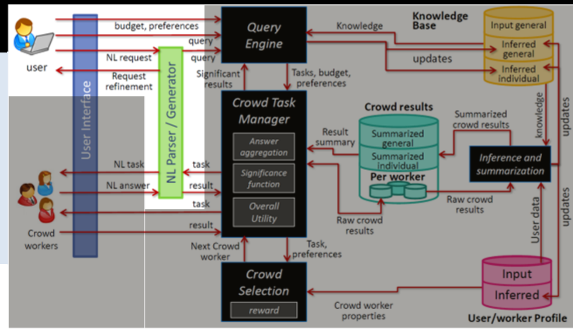
Using: Value, Dismantling, and Example crowd questions

Output:

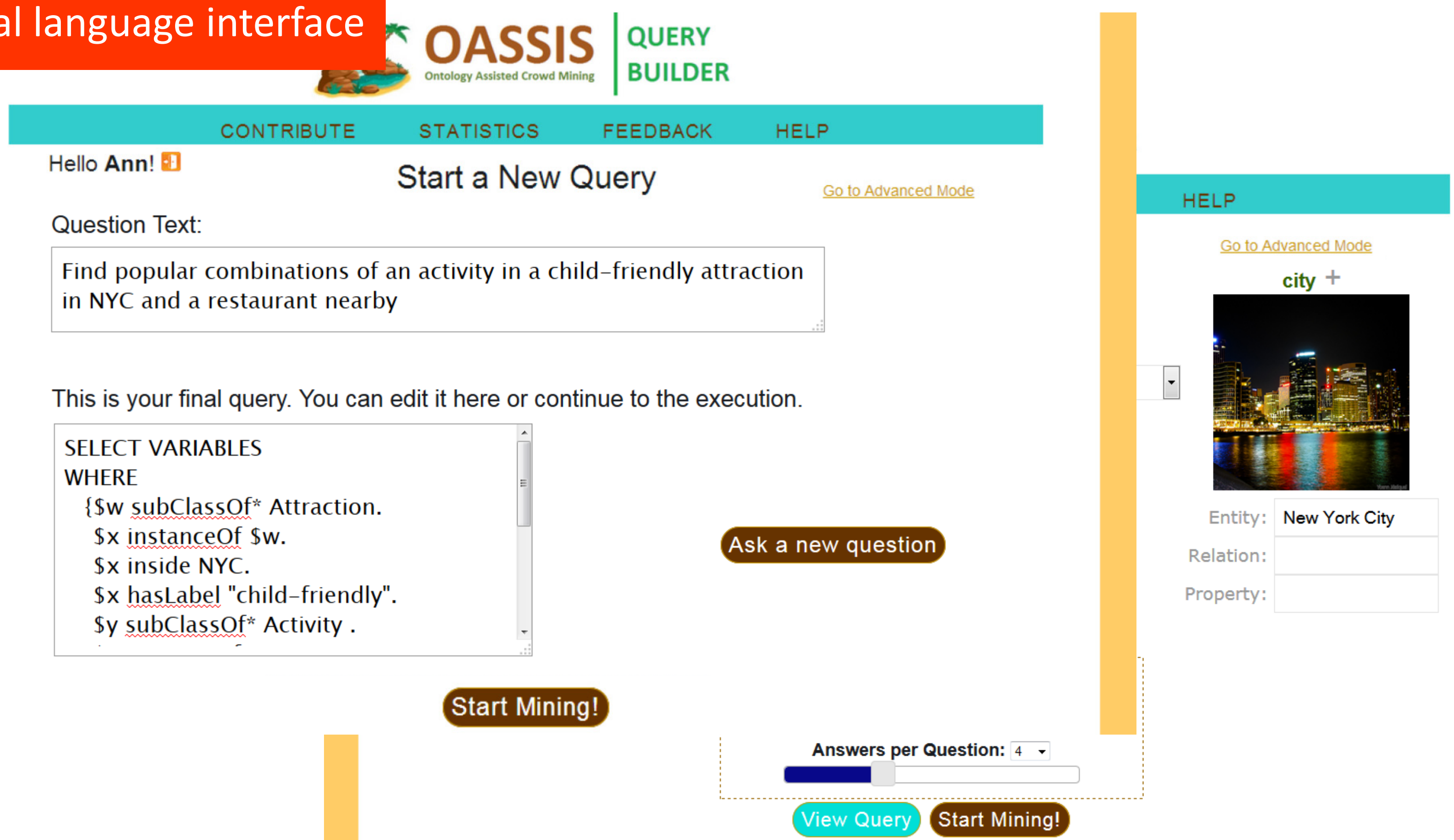
1. How many questions to ask on each att. (a Budget distribution)
2. How to compose the answers (a Linear regression)

- $\mathrm{BMI}^{(20)}$
- $0.7\mathrm{BMI}^{(10)} + 0.1\mathrm{Weight}^{(6)} + 6.5\mathrm{Fat}^{(4)} + 4.06$
- $0.2\mathrm{BMI}^{(4)} + 9.5\mathrm{Heavy}^{(3)} + 0.2\mathrm{Weight}^{(2)} + 0.4\mathrm{GoodBuilt}^{(2)} + 4.9\mathrm{Over200Pounds}^{(4)} - 0.3\mathrm{FairLooking}^{(1)} - 2.7\mathrm{GoodFacialFeatures}^{(1)} - 0.2\mathrm{GoodPhysicalFeatures}^{(1)} + 0.6\mathrm{HasWork}^{(1)} - 0.1\mathrm{WorksOut}^{(1)} + 12.6$

Formulating a Query?



Natural language interface



The Case for Natural Language (NL)

- General and individual knowledge needs can be mixed in an intricate manner in NL

> "What are the $\boxed{\text{most interesting}}$ $\boxed{\text{places near Forest Hotel, Buffalo,}}$ we $\boxed{\text{should visit in the fall?}}$"

- **Our goal:**
  - identifying the knowledge needs of each type
  - and translating them into formal queries

Challenges

- **Distinguishing** general from individual expressions in the question

Opinion Mining tools can detect some individual expressions (opinions, preferences) but not all (habits and practices)

Matching to a knowledge base is insufficient to detect general expressions, since knowledge bases are dirty and incomplete

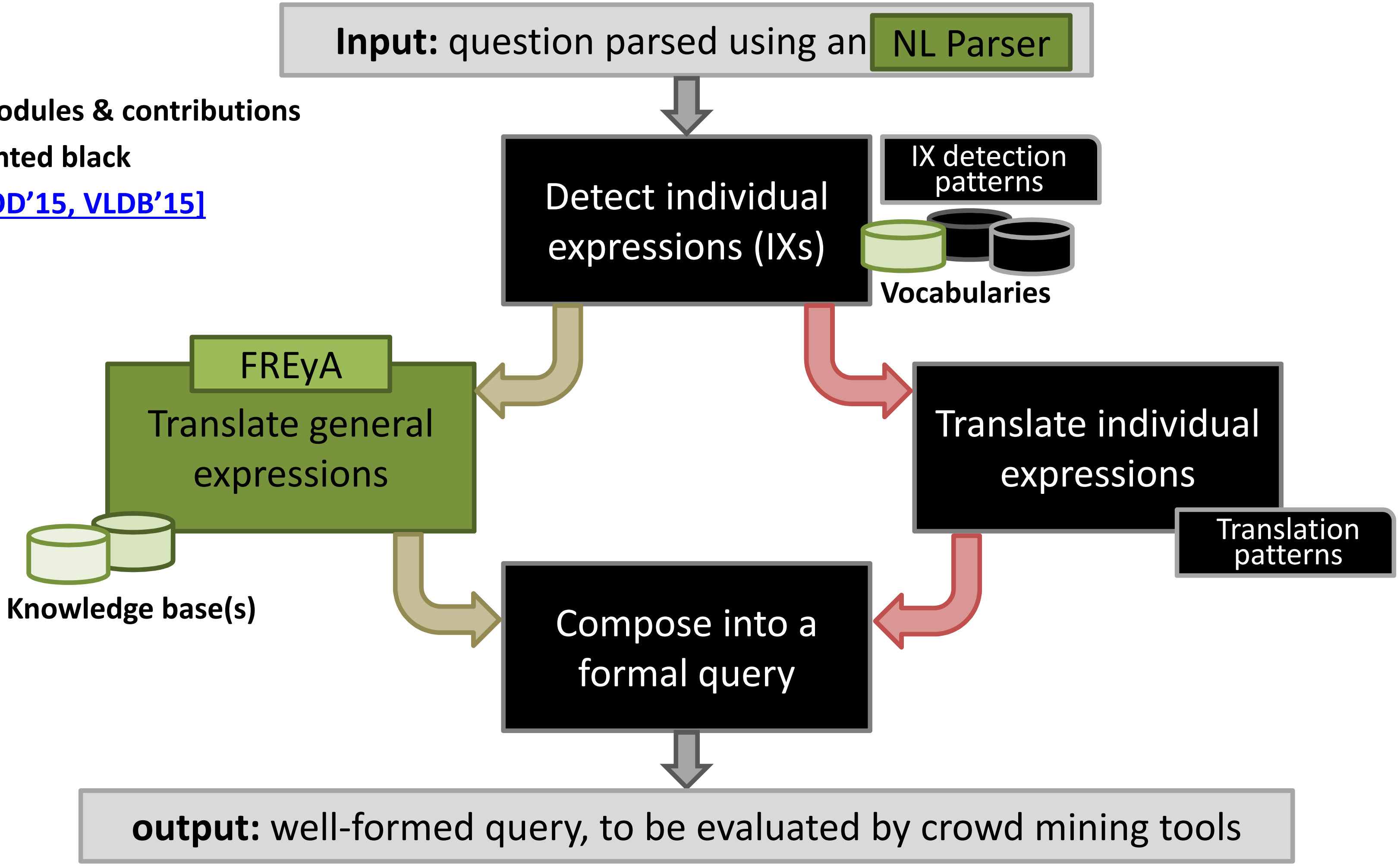
- **Translating** each expression accordingly

Existing NL-to-query translation tools rely on a knowledge base and are thus irrelevant for individual expressions

- **Integrating** the results into a well-formed query

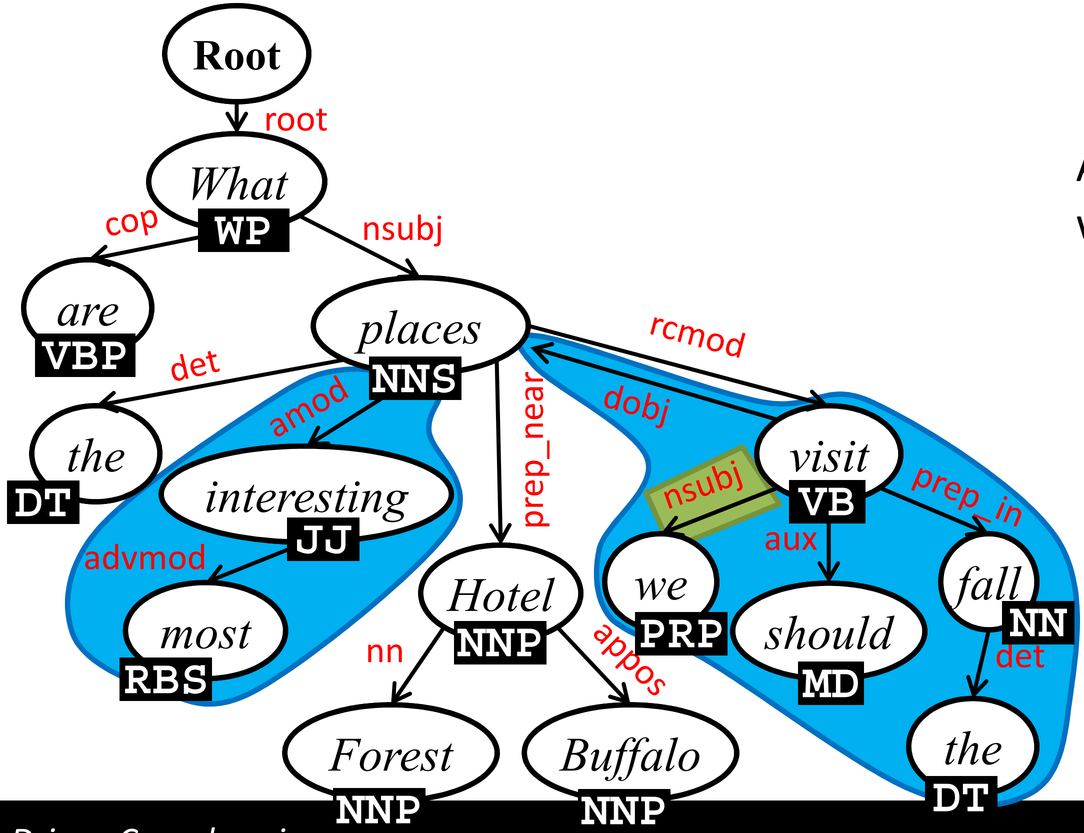
Solution Outline – A modular Framework

New modules & contributions are painted black
[SIGMOD'15, VLDB'15]



The IX detection Module

- The parsed NL sentence can be represented as a directed, labeled graph
- We use SPARQL-like selection patterns and vocabularies to detect Ixs within the graph, and then other patterns to select the full IXs



A (simplified) example: a verb (VB) whose subject (nsubj) is individual

```
$x nsubj $y
filter($y in V_participant &&
      (POS($x) = "VBP" ||
       POS($x) = "VB"))
```

Detection & Translation Patterns

- IX detection
  - Lexical Individuality (e.g. “interesting place” vs. “northern place”)
  - Participant individuality (e.g. “we visit” vs. “Tim Duncan visits”)
  - Syntactic individuality (e.g. “Tim Duncan should play for the NY Knicks“)

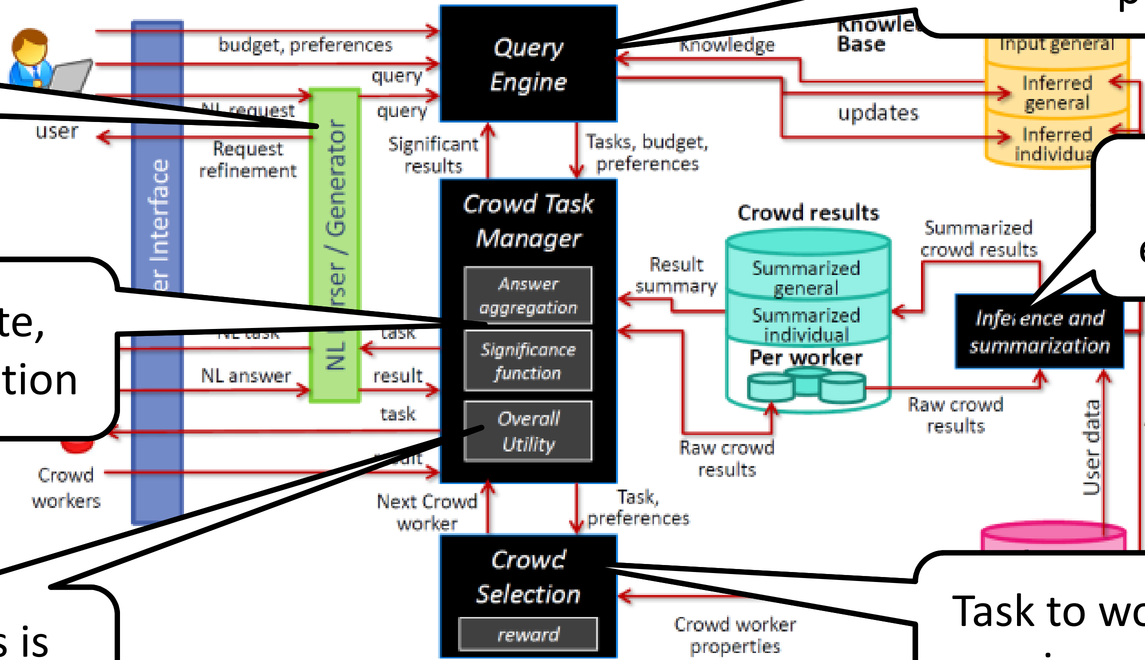
- Query creation
  - NL to OASSIS-QL mapping

{$x amod $y} -> {tran($x) hasLabel “tran($y)”}

6 hasLabel "interesting"

Before We Conclude

Other crowdsourcing systems can be put in terms of the architecture for comparing and identifying possible extensions



Declarative crowdsourcing platforms

NL to query translators

Crowdsourced entity resolution

Majority vote, costume function

\# questions is fixed or bounded

Task to worker assignment

Summary

The crowd is an incredible resource!

"Computers are useless, they can only give you answers"

- Pablo Picasso

But, as it seems, they can also ask us questions!

Many challenges:

- Almost no theory (and when exists, too "clean")
- (very) interactive computation
- A huge amount of data
- Varying quality and trust

Thanks

Antoine Amerilli, Yael Amsterdamer, Moria Ben-Ner, Rubi Boim, Susan Davidson, Ohad Greenshpan, Benoit Gross, Yael Grossman, Ana Itin, Ezra Levin, Ilia Lotosh, Slava Novgordov, Neoklis Polyzotis, Sudeepa Roy, Pierre Senellart, Amit Somech, Wang-Chiew Tan, Brit Youngmann, …