Human Computation

Core Research Questions and State of the Art

part 1

Edith Law
Carnegie Mellon University

Human Computation

in a nutshell

Human Computation

in a nutshell

"Some problems are hard, even for the most sophisticated AI algorithms."

Human Computation

in a nutshell

"Some problems are hard, even for the most sophisticated AI algorithms."

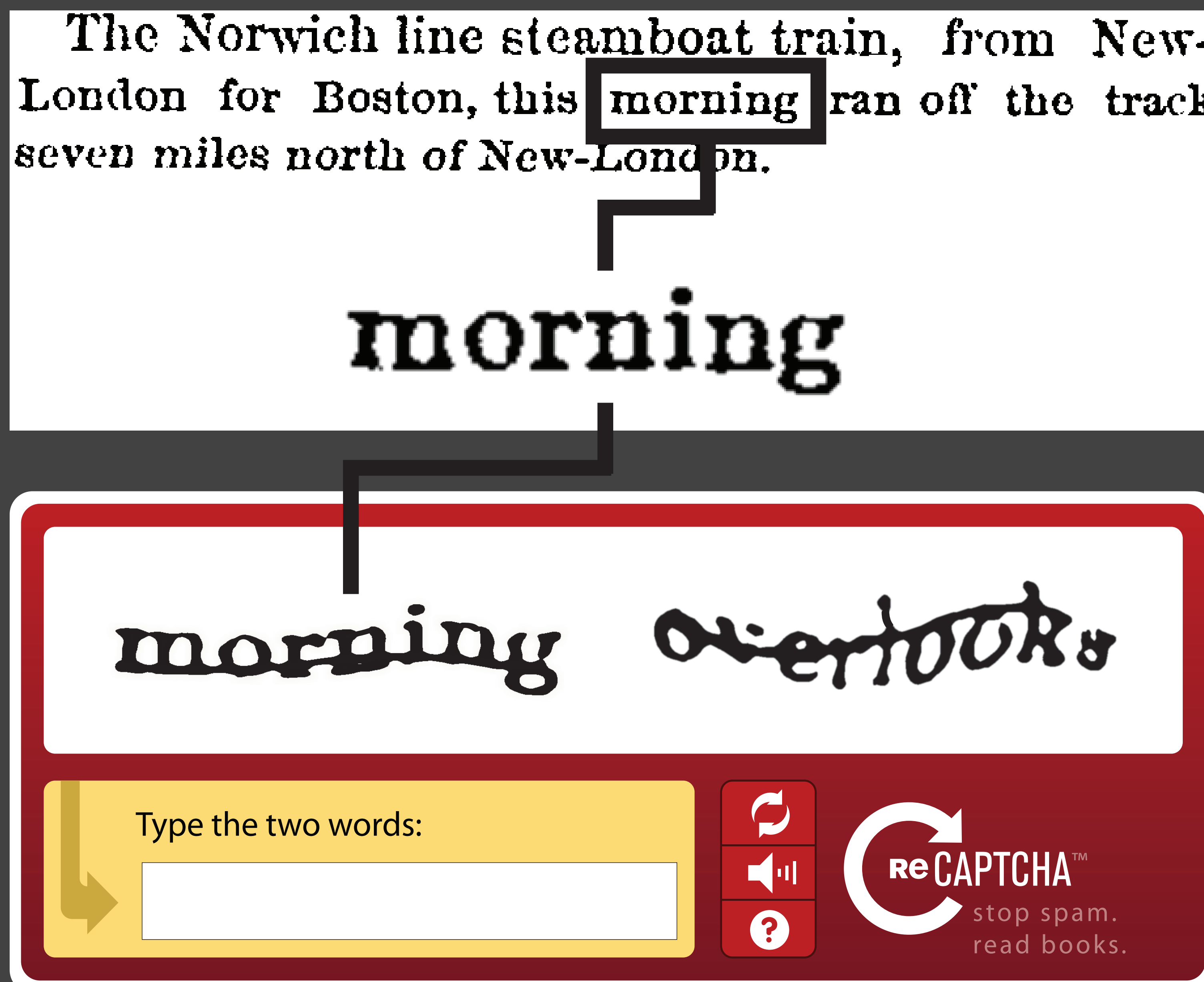
"Let humans solve it ... "

Human Computation

you are a human computer

Human Computation

you are a human computer



Human Computation is an old idea.

The Past

a very brief history

1800 1850 1900 1950

“When Computers were Human” by David A. Grier

The Past

a very brief history

Halley Comet
1750's

source: the Yerkes Observatory

1800 1850 1900 1950

“When Computers were Human” by David A. Grier

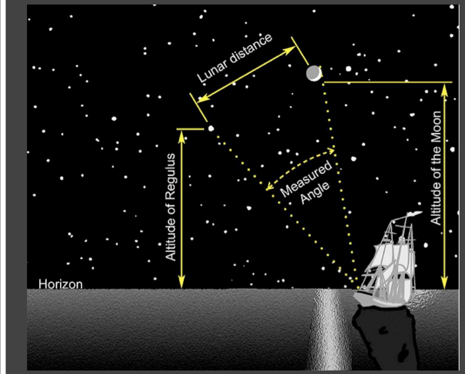
The Past

a very brief history

Halley Comet
1750's

source: the Yerkes Observatory

Nautical Almanac
1770's



1800 1850 1900 1950

"When Computers were Human" by David A. Grier

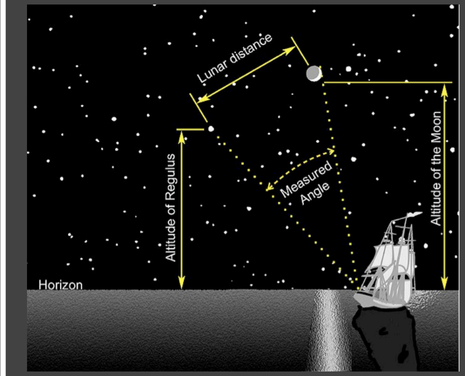
The Past

a very brief history

Halley Comet
1750's

source: the Yerkes Observatory

Nautical Almanac
1770's



Math Tables
Project
1930's

courtesy: David A. Grier

1800 1850 1900 1950

"When Computers were Human" by David A. Grier

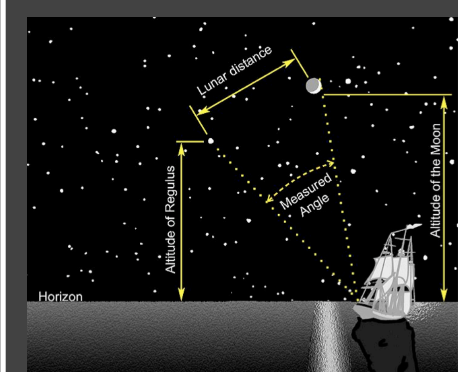
The Past

a very brief history

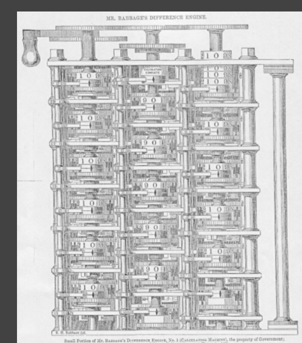
Halley Comet
1750's

source: the Yerkes Observatory

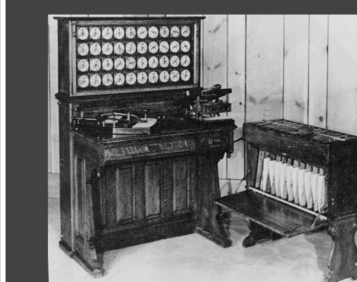
Nautical Almanac
1770's



Babbage
Difference Machine
1820's



Hollerith
Machine
1890's



source: IBM

Math Tables
Project
1930's

courtesy: David A. Grier

1800 1850 1900 1950

"When Computers were Human" by David A. Grier

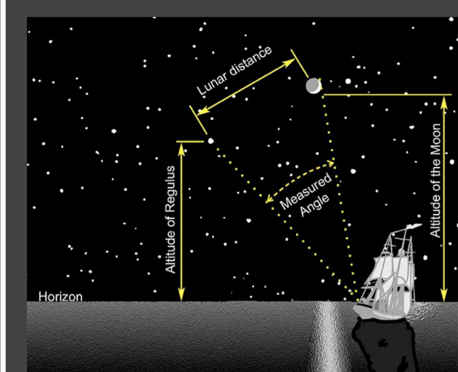
The Past

a very brief history

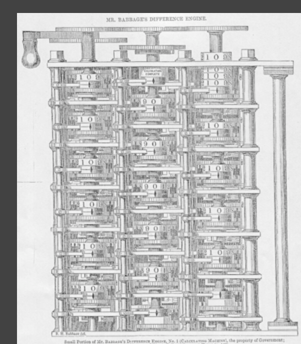
Halley Comet
1750's

source: the Yerkes Observatory

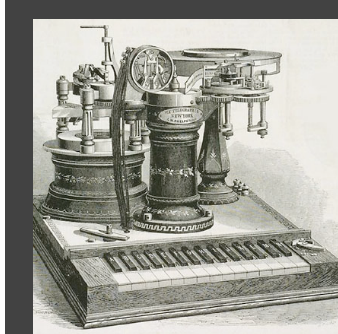
Nautical Almanac
1770's



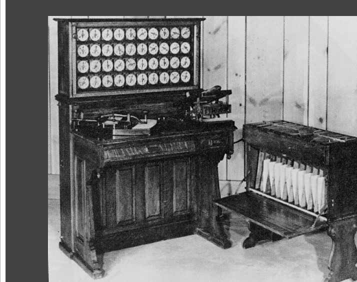
Babbage
Difference Machine
1820's



Telegraph &
Weather
1850's



Hollerith
Machine
1890's



source: IBM

Math Tables
Project
1930's

courtesy: David A. Grier

1800 1850 1900 1950

"When Computers were Human" by David A. Grier

The Web changed everything.

The Present

scale

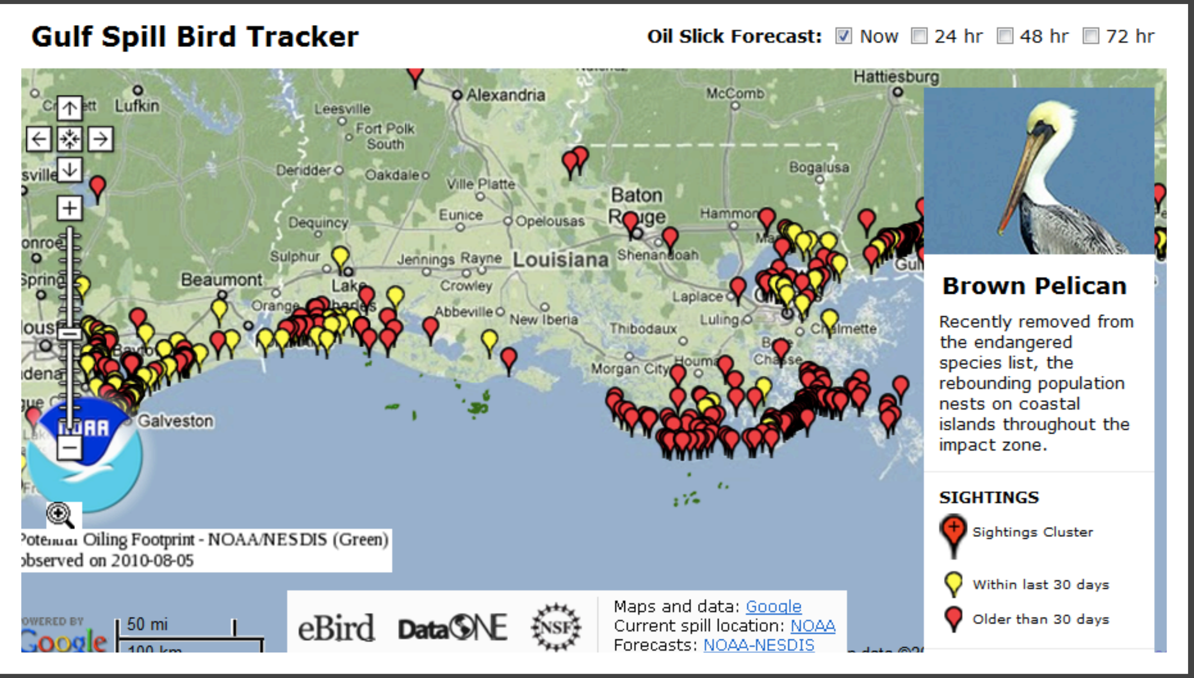
(von Ahn and Dabbish, 2004)

The Present

reach

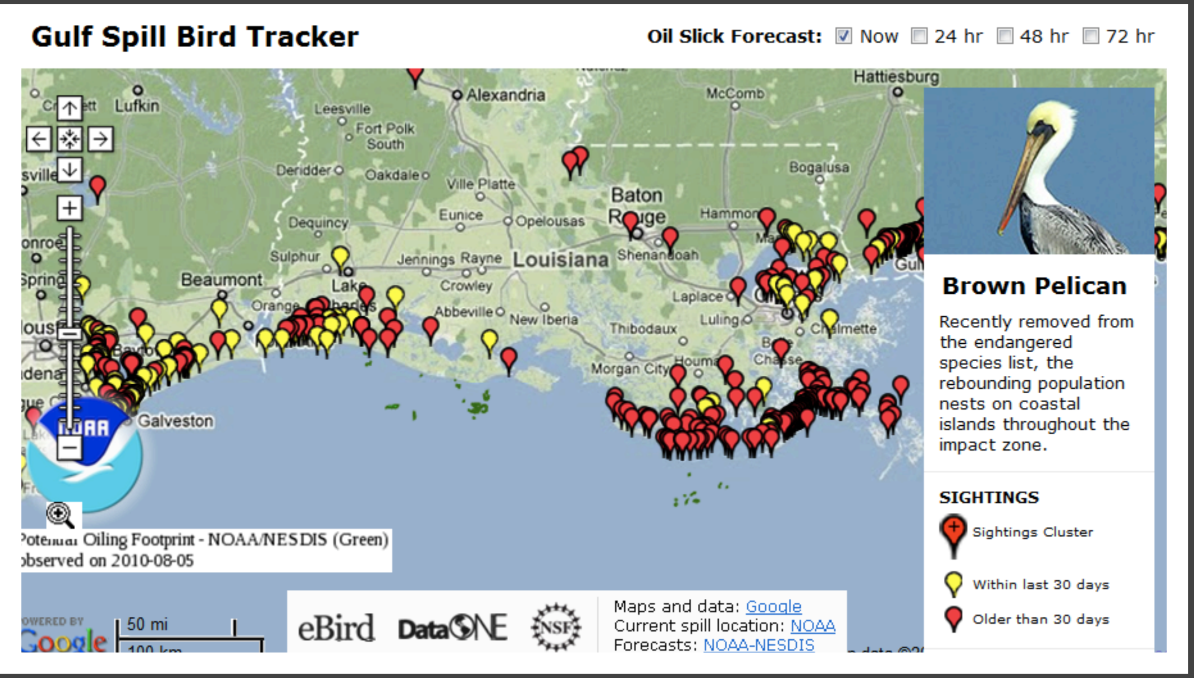
The Present

reach



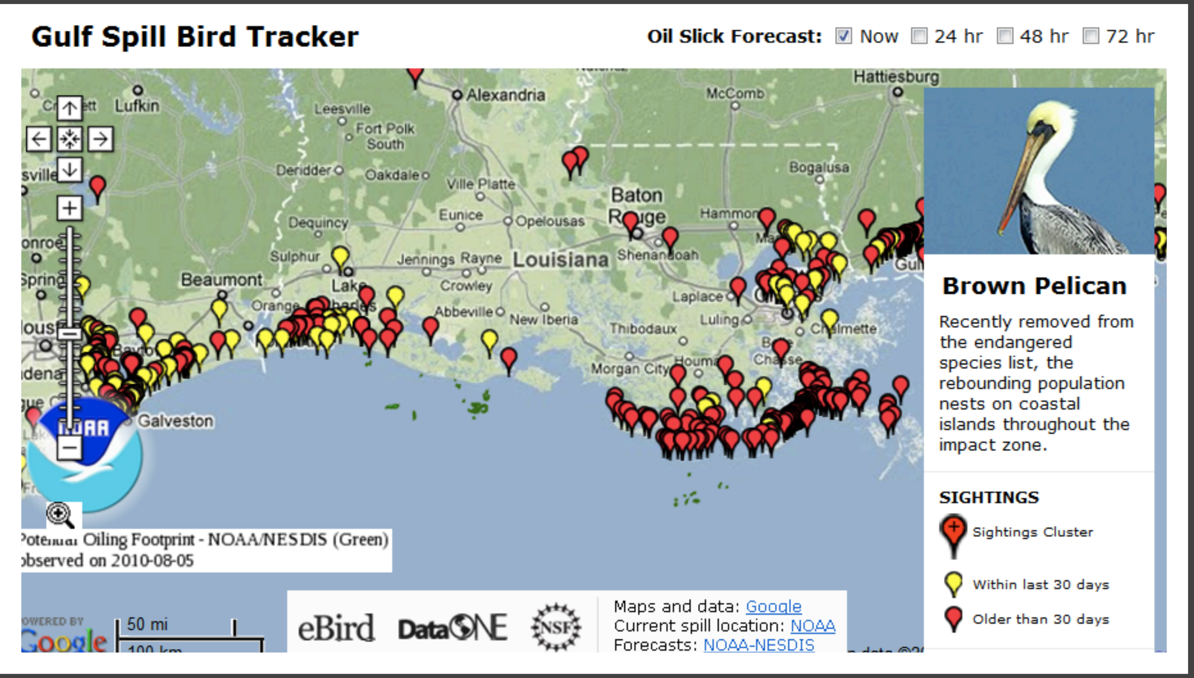
The Present

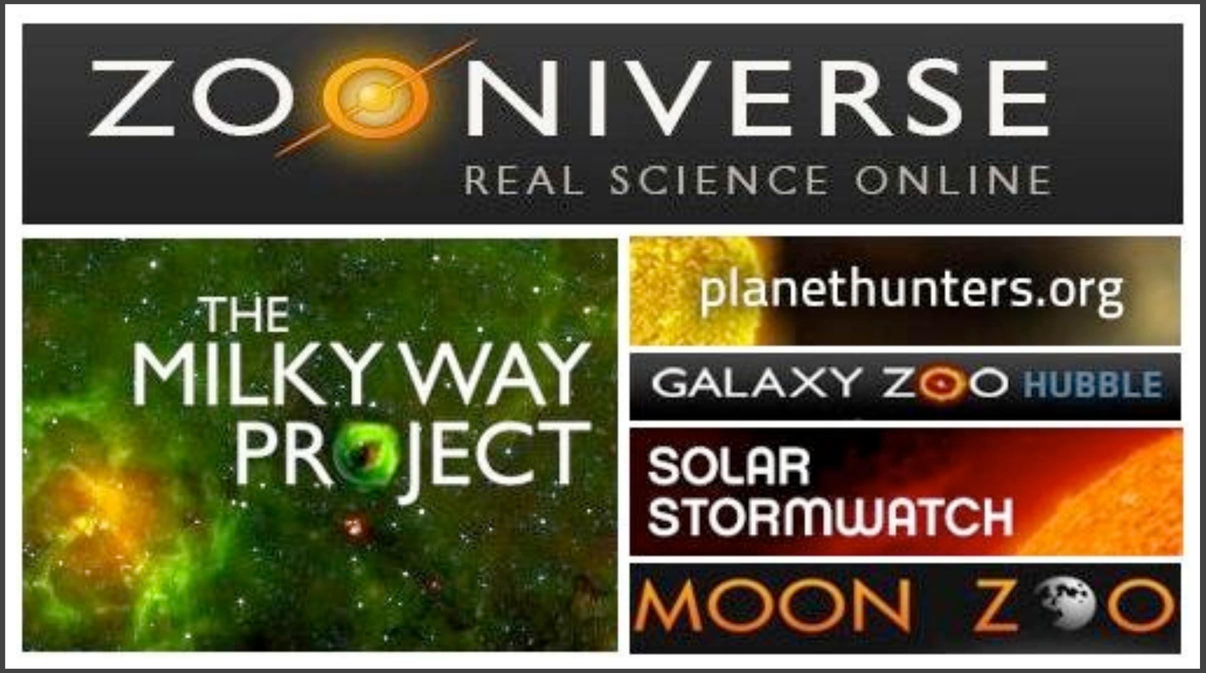
reach



The Present

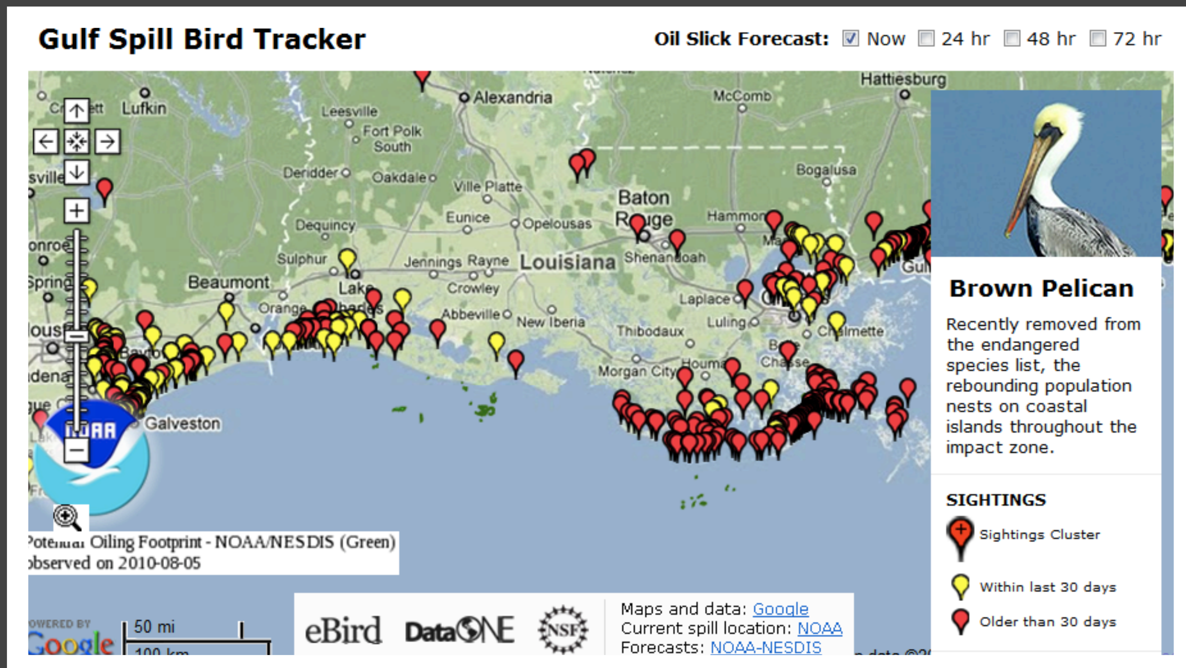
reach

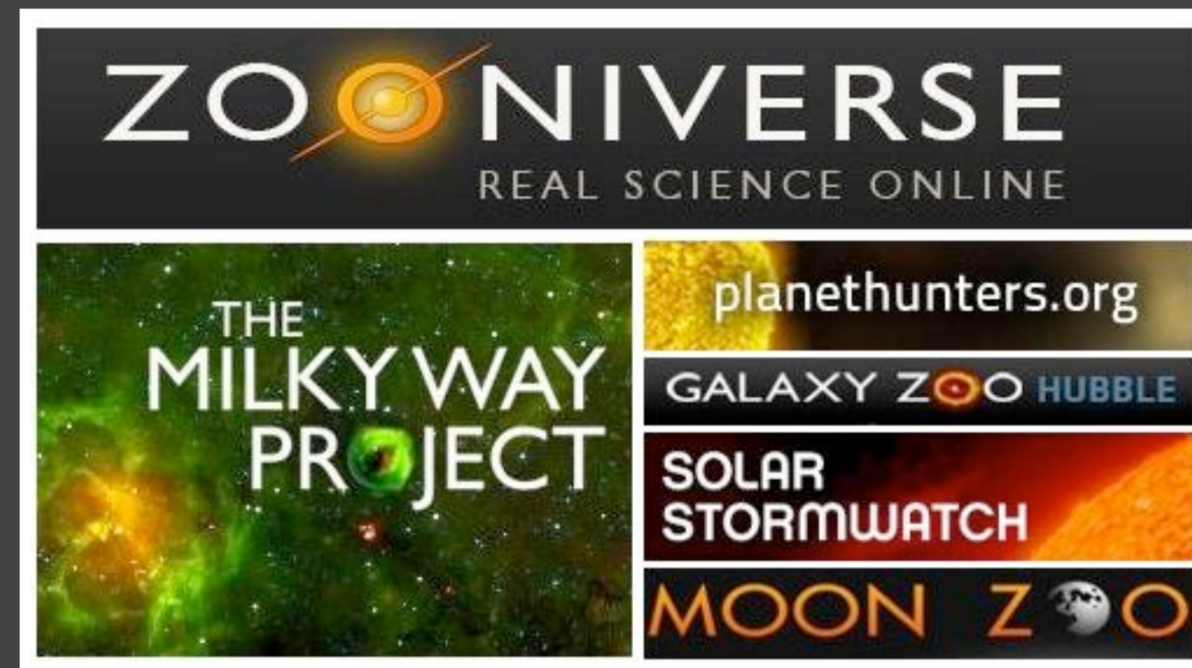




The Present

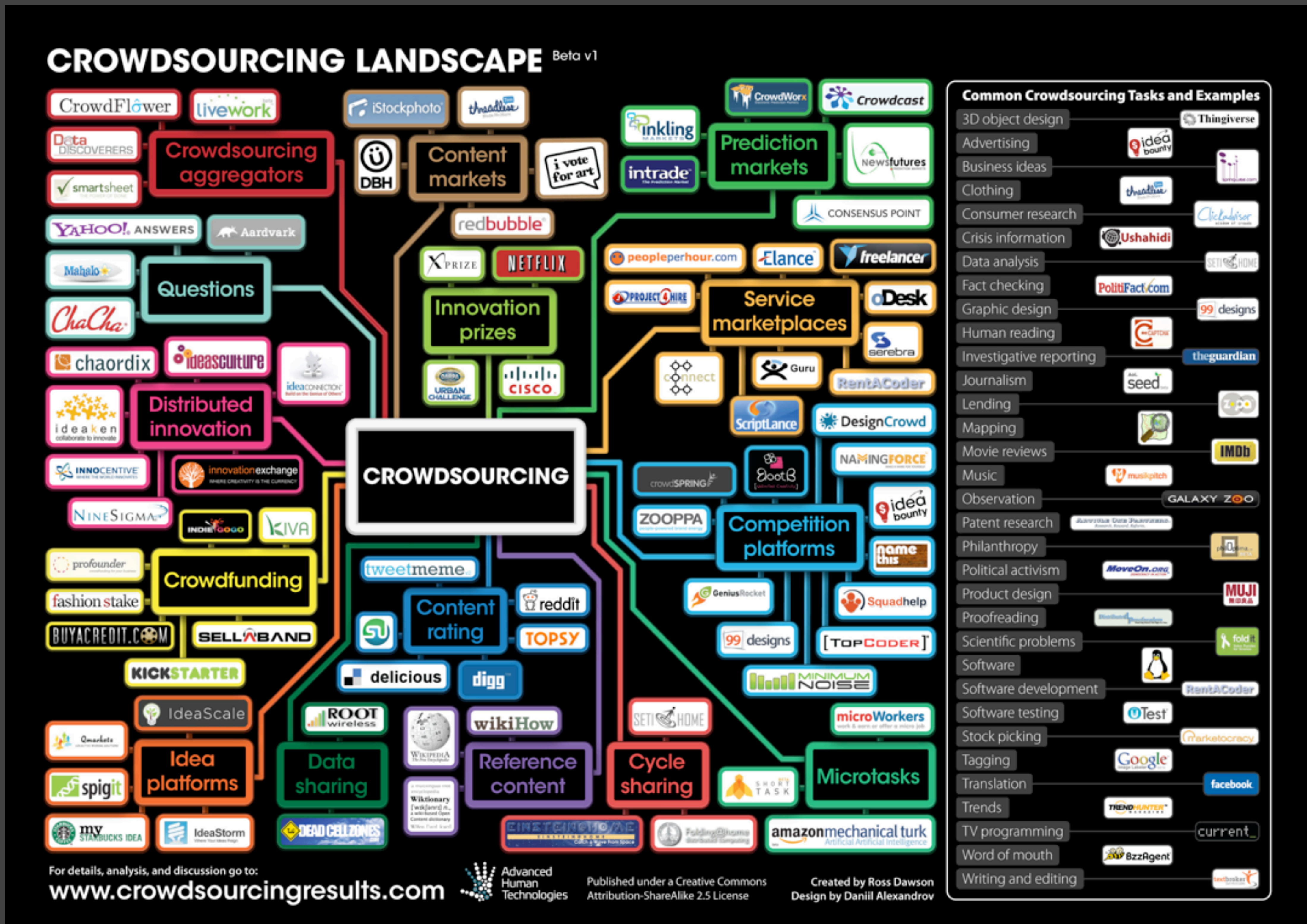
reach





The Present

pervasiveness



Human Computation

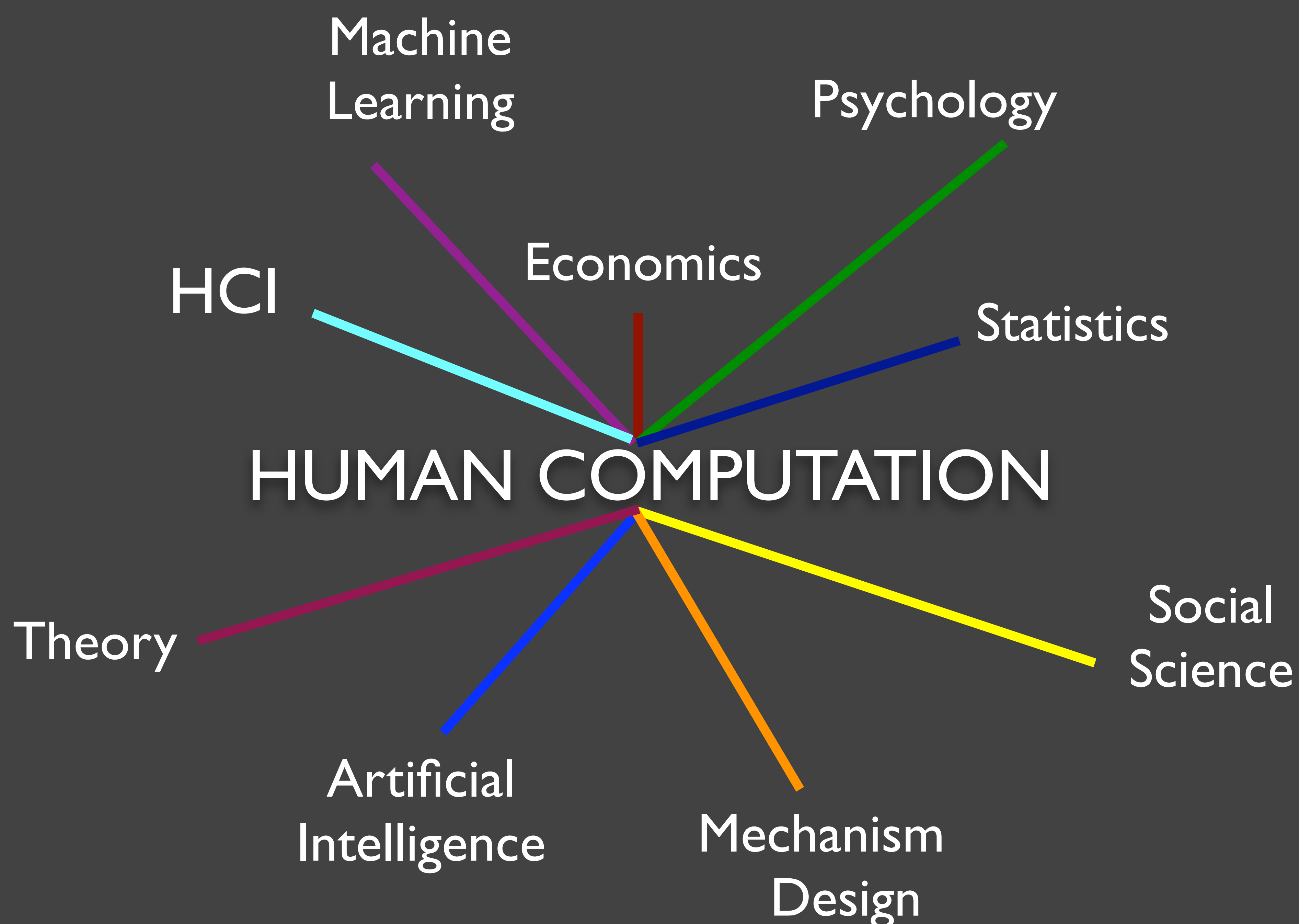
a growing field

1st Human Computation Workshop	KDD 2009	WORKSHOPS
Crowdsourcing for Search Evaluation	SIGIR 2010	
2nd Human Computation Workshop	KDD 2010	
Advancing Computer Vision with Humans in the Loop	CVPR 2010	
Creating Speech and Language Data with Amazon's Mechanical Turk	NAACL 2010	
Computational Social Science and Wisdom of the Crowds	NIPS 2010	
Workshop on Ubiquitous Crowdsourincg	Ubicomp 2010	
Enterprise Crowdsourcing Workshop	ICWE 2010	
Collaborative Translation Technology, Crowdsourcing and the Translator	AMTA 2010	
Crowdsourcing for Search and Data Mining	WSDM 2010	
Workshop on Crowdsourcing for Information Retrieval	SIGIR 2011	
Workshop on Social Computing and User Generated Content	EC 2011	
Workshop on Crowdsourcing and Human Computation	CHI 2011	
3rd Human Computation Workshop	AAAI 2011	

Mechanical Turk for Computer Vision	CVPR 2010	TUTORIALS
Crowdsourcing for Relevance Evaluation	ECIR 2010	
Managing Crowdsourced Human Computation	WWW 2011	
Crowdsourcing 101: Putting the WSDM of Crowds to Work for You	WSDM 2011	
Crowdsourcing Applications and Platforms	VLDB 2011	
Crowdsourcing for Information Retrieval: : Principles, Methods and Appplications	SIGIR 2011	
Quality Crowdsourcing for Human Computer Interaction Research	HCIC 2011	
Crowdsourcing for Fun and Profit	CrowdConf 2011	
Human Computation: Core Research Questions and State of the Art	AAAI 2011	

Human Computation

multi-disciplinary



Tutorial

with a purpose

Introduce a framework for human computation with a set of concepts, core research questions, existing work and open problems.

Tutorial
with a purpose

Introduce a framework for human computation with a set of concepts, core research questions, existing work and open problems.

Research Opportunities indicated by

Tutorial

a rough schedule

ALGORITHM

part 1

2:00-3:30
Edith Law

DESIGN

part 2

4:00-5:30
Luis von Ahn

I

A FRAMEWORK FOR HUMAN COMPUTATION

Concepts • Scope

CONCEPTS

Computation

a general definition

The process of mapping input to output.

Computational Problems

examples

multiplication	two numbers ➨	product
sorting	set of objects ➨	set of objects sorted
medical diagnosis	x-ray, lab tests ➨	diagnosis
object recognition	image ➨	tag
translation	source sentence ➨	target sentence
editing	text ➨	corrected text
planning	goal, constraints ➨	sequence of actions

Human Computation
a general definition

Computation that is carried out by a human.

Human Computation
problem statement

Human Computation
problem statement

Given a computational problem, design a solution using human computers and automated computers.

Related Concepts

definitions

Related Concepts
definitions

COLLECTIVE INTELLIGENCE

The shared or group intelligence that emerges from the collaboration and competition of many individuals (bacteria, animals, humans, computer agents).

Related Concepts
definitions

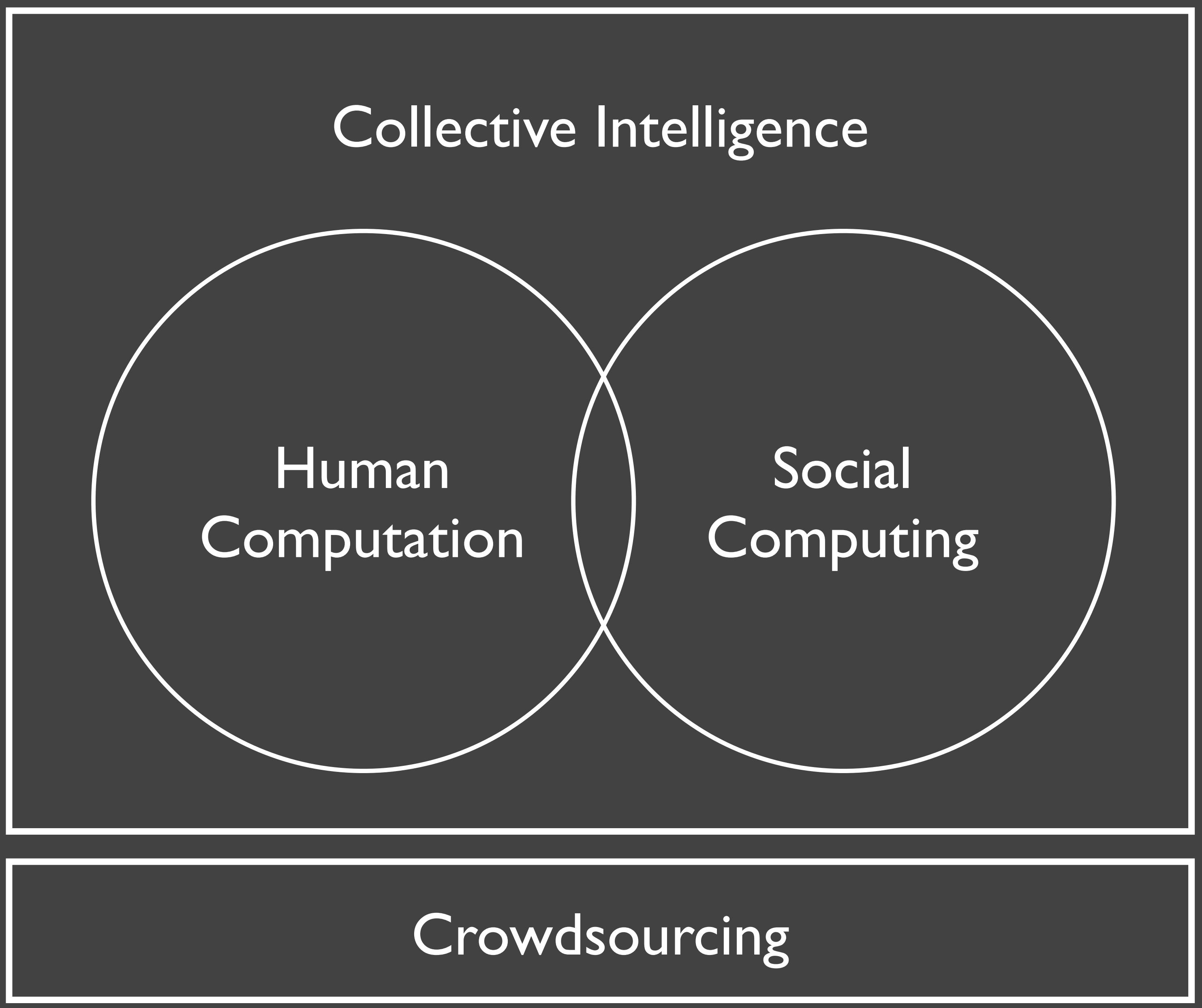
COLLECTIVE INTELLIGENCE	The shared or group intelligence that emerges from the collaboration and competition of many individuals (bacteria, animals, humans, computer agents).
SOCIAL COMPUTING	Technology for supporting social behavior and interactions (e.g., blog, email, Instant messaging) or group computation (e.g., collaborative filtering, auctions, prediction markets).

Related Concepts
definitions

COLLECTIVE INTELLIGENCE	The shared or group intelligence that emerges from the collaboration and competition of many individuals (bacteria, animals, humans, computer agents).
SOCIAL COMPUTING	Technology for supporting social behavior and interactions (e.g., blog, email, Instant messaging) or group computation (e.g., collaborative filtering, auctions, prediction markets).
CROWDSOURCING	Outsourcing tasks through an open call.

Related Concepts

boundaries



Three Distinguishing Features

of human computation

Three Distinguishing Features
of human computation

Three Distinguishing Features

of human computation

"Human" In The Loop

not bacteria, not ants, not fish.

Three Distinguishing Features

of human computation

“Human” In The Loop

not bacteria, not ants, not fish.

Three Distinguishing Features

of human computation

“Human” In The Loop

not bacteria, not ants, not fish.

Conscious Effort

humans are actively computing something, not merely carrier of sensors and computational devices.

Three Distinguishing Features

of human computation

“Human” In The Loop

not bacteria, not ants, not fish.

Conscious Effort

humans are actively computing something, not merely carrier of sensors and computational devices.

Three Distinguishing Features
of human computation

"Human" In The Loop
not bacteria, not ants, not fish.

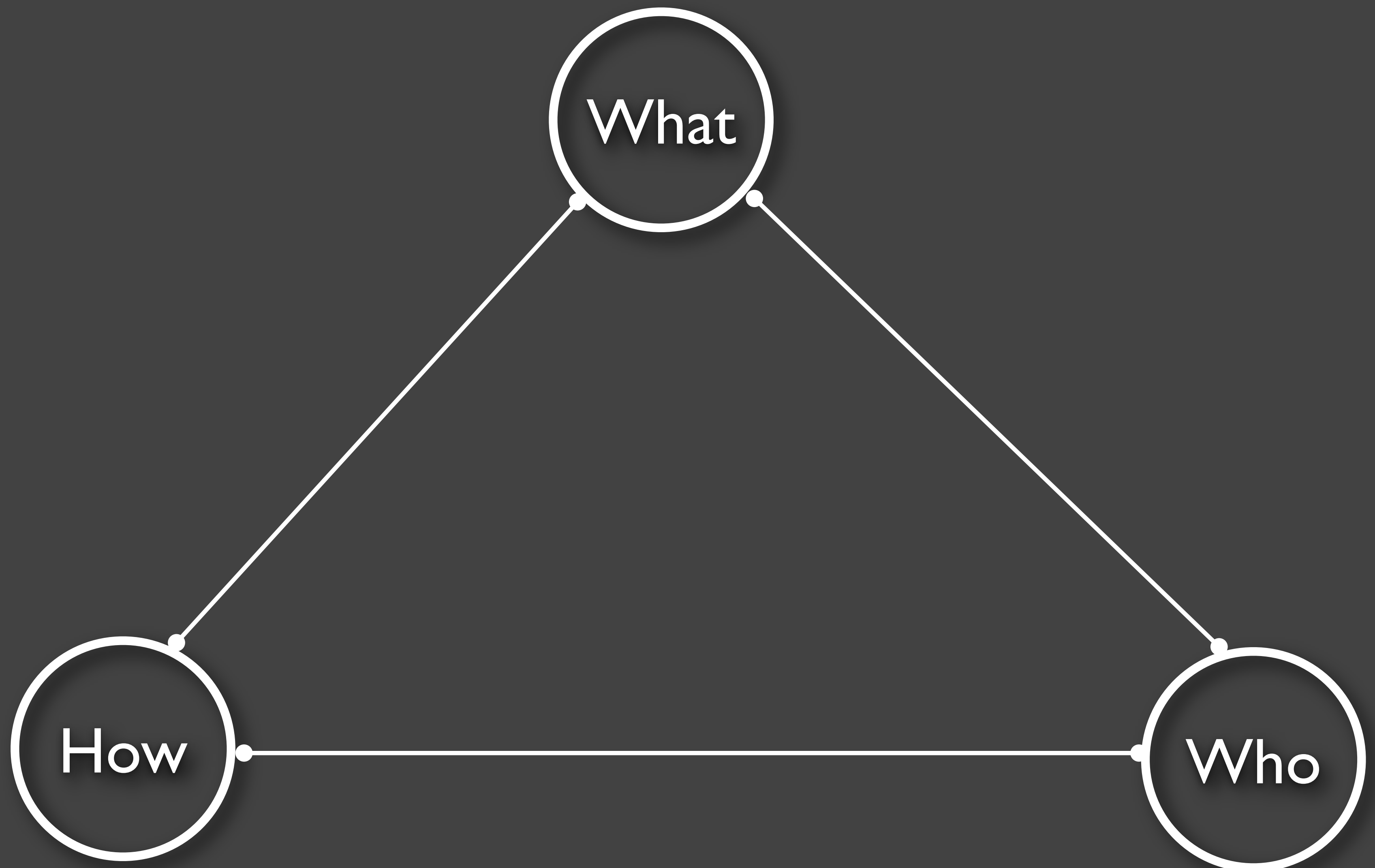
Conscious Effort
humans are actively computing something, not merely carrier of sensors and computational devices.

Explicit Control
the outcome of the computation is determined by an algorithm, and not the natural dynamics of the crowd.

SCOPE

Core Research Questions

"what", "who", "how"



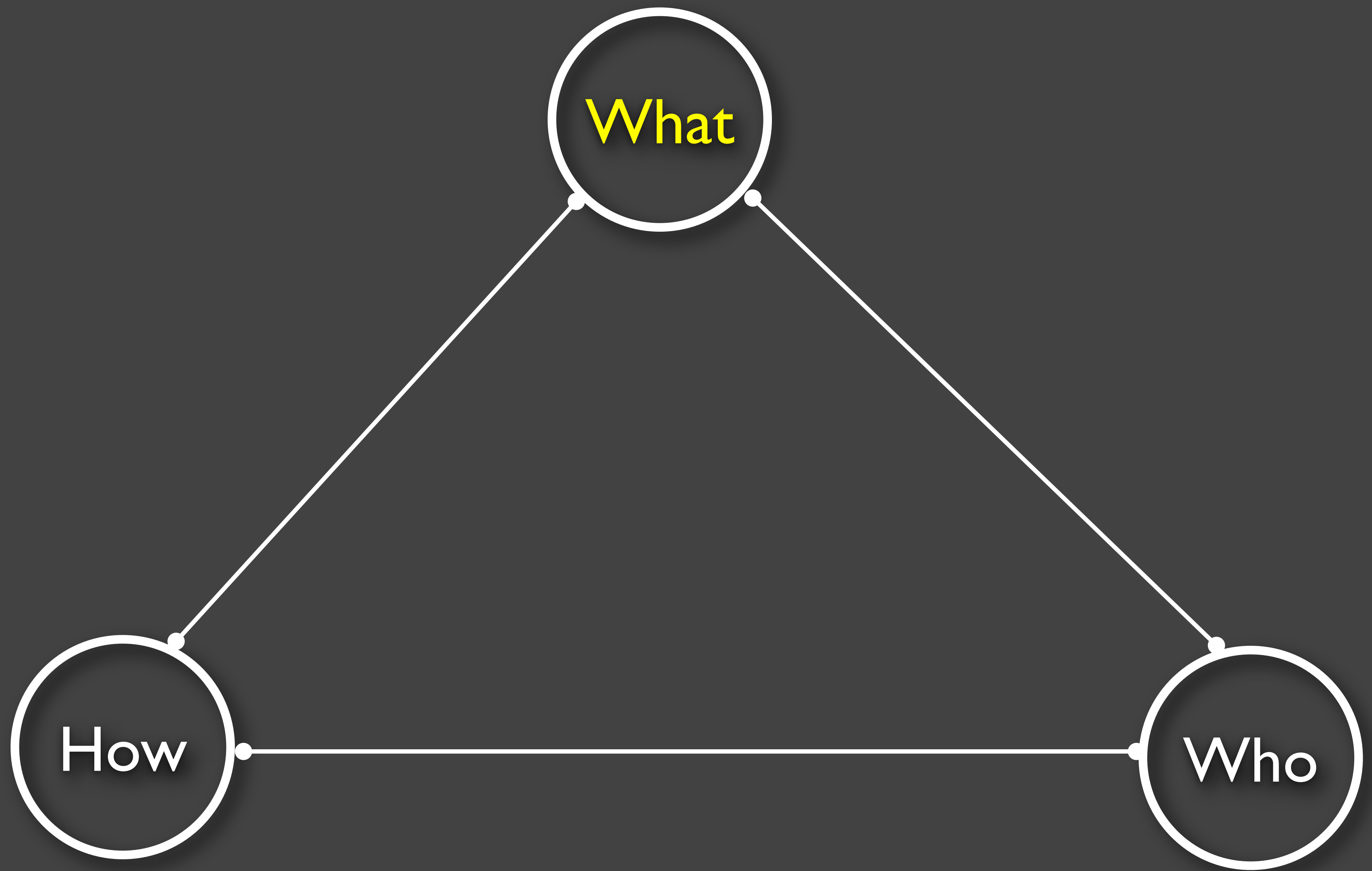
Core Research Questions

from the definition

Given a computational problem, design a solution using human computers and automated computers.

Core Research Questions

“what”, “who”, “how”



Core Research Questions

from the definition

"How hard is the problem? Is it efficiently solvable?"

↓

Given a computational problem, design a solution using human computers and automated computers.

Core Research Questions

from the definition

"Is the human computation algorithm correct and efficient?"

↓

Given a computational problem, design a solution using human computers and automated computers.

Core Research Questions

from the definition

Given a computational problem, design a solution using human computers and automated computers.

↑

"How do we aggregate the outputs of many human computers?"

Core Research Questions

from the definition

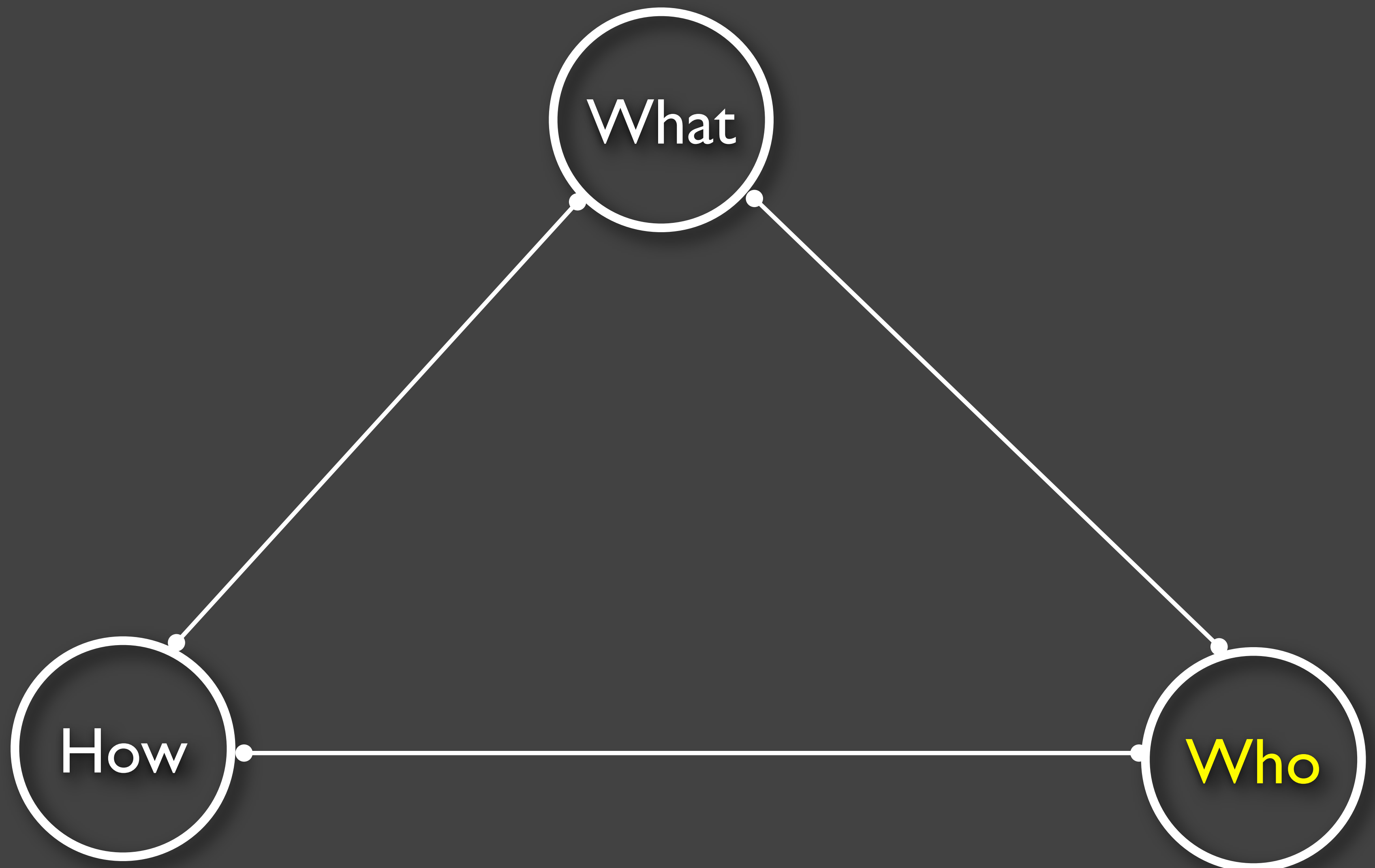
Given a computational problem, design a solution using human computers and automated computers.

↑

"How to make the tradeoff between human versus machine?"

Core Research Questions

"what", "who", "how"



Core Research Questions

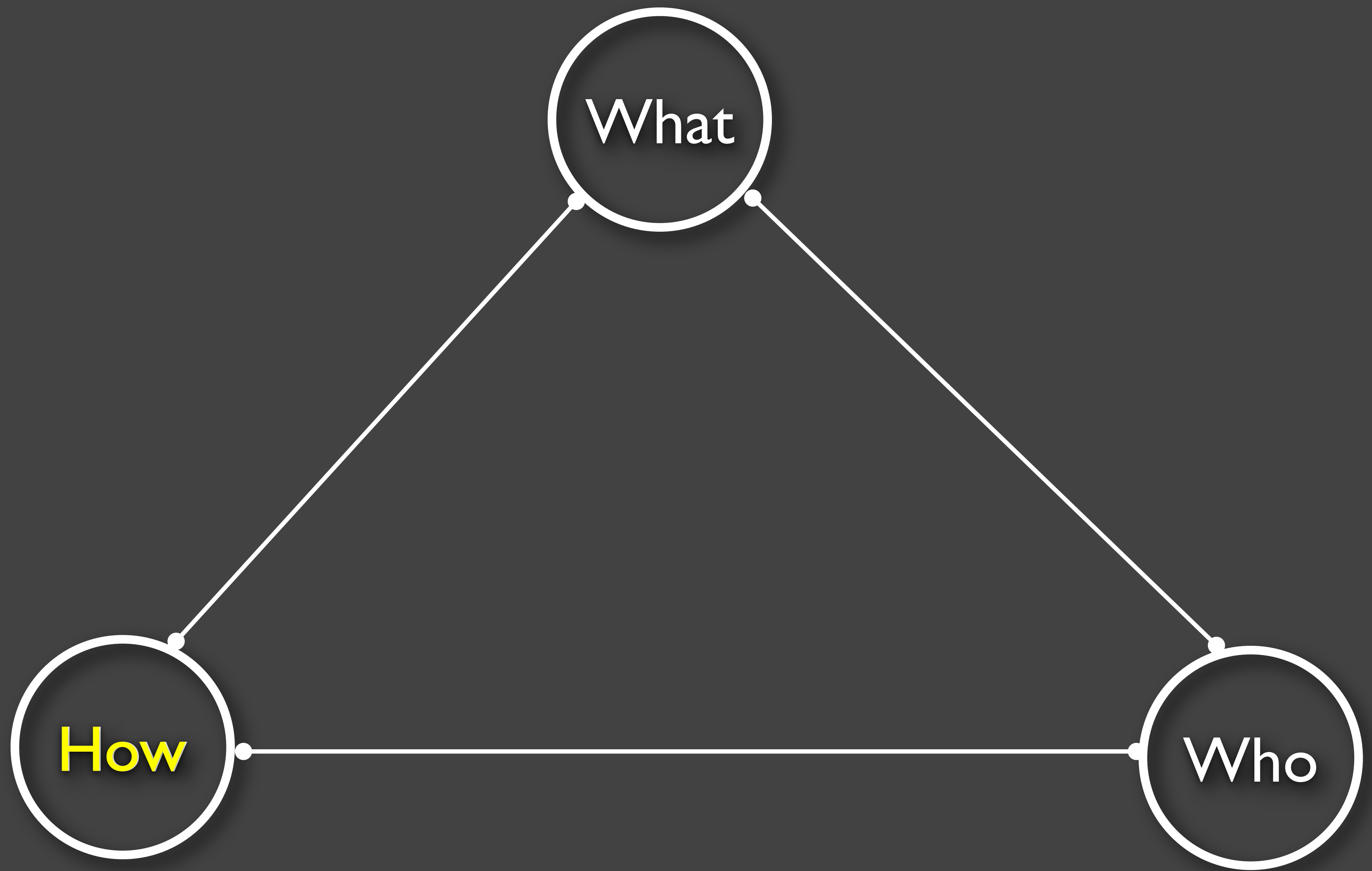
from the definition

“ To whom do we route each task, and how?”

Given a computational problem, design a solution using human computers and automated computers.

Core Research Questions

"what", "who", "how"



Core Research Questions

from the definition

Given a computational problem, design a solution using human computers and automated computers.

↑

"How to design tasks, motivate participation and incentivize truthful outputs?"

Core Research Questions

from the definition

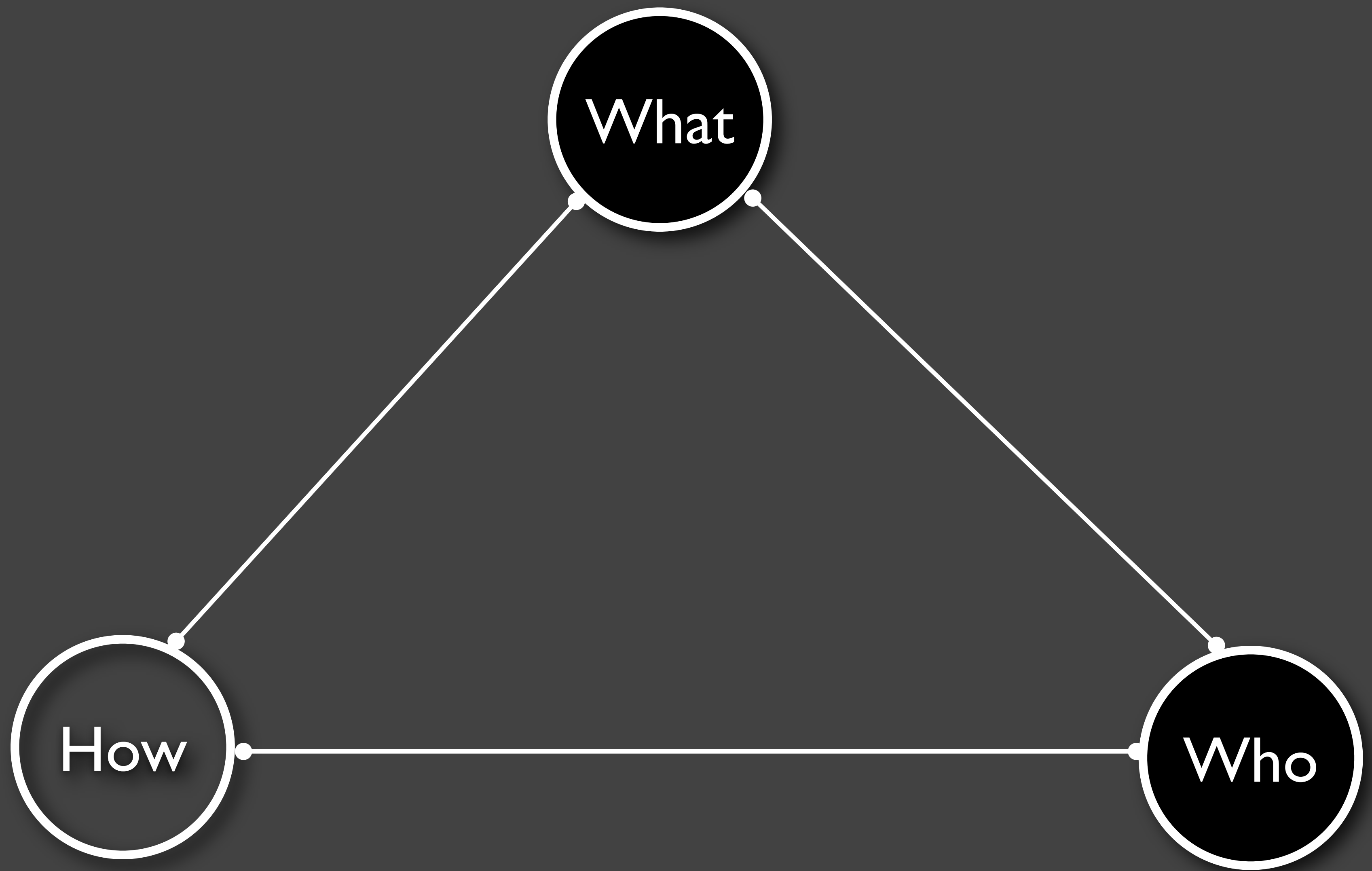
"How to meet the needs and wants of the requesters?"

↓

Given a computational problem, design a solution using human computers and automated computers.

Core Research Questions

“what”, “who”, “how”



II

HUMAN COMPUTATION ALGORITHMS

Definition • Properties

DEFINITION

What are algorithms?

Algorithms

an example

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return (x < pivot)
```

Algorithms

an example

Inputs

↓

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return (x < pivot)
```

Algorithms

an example

Inputs

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return (x < pivot)
```

Outputs

Algorithms

an example

Inputs

Precisely Defined Steps

Outputs

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return (x < pivot)
```

Algorithms

an example

Inputs

Termination Condition

Precisely Defined Steps

Outputs

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return (x < pivot)
```

Algorithms

an example

Inputs

Termination Condition

Precisely Defined Steps

Outputs

Sufficiently Basic Operations

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return (x < pivot)
```

Algorithms

a definition

“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973

Algorithms

a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- Knuth, 1973

Algorithms

a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- Knuth, 1973

Algorithms

a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- Knuth, 1973

Algorithms

a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- Knuth, 1973

Algorithms

a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- *Knuth, 1973*

Algorithms

a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- Knuth, 1973

Algorithms

a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- Knuth, 1973

Algorithms
a definition

"An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness."

- *Knuth, 1973*

Algorithms

a definition

“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973

Algorithms

a definition

“An algorithm is a finite set of rules which gives a sequence of operations for solving a specific type of problem, with five important properties:

Input, Output, Finiteness, Definiteness, Effectiveness.”

- Knuth, 1973

What are human computation algorithms?

Human Computation Algorithms

human-driven operation

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return (x < pivot)
```

Human Computation Algorithms

human-driven operation

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);
```

Human Computation Algorithms

human-driven operation

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

Human Computation Algorithms

human-driven operation

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

Human Computation Algorithms

human-driven operation

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

Mechanical Turk Task

Instructions

You are shown two images. You must select the image that is more indicative of suspicious activities.

Task

Imagine that you are a security guard and you are monitoring two places. Someone informed you that there are suspicious activities in one of the places, but you were not told which one. Which place will you attend to?

Submit

Human Computation Algorithms

human-driven operation

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

Mechanical Turk Task

Instructions

You are shown two images. You must select the image that is more indicative of suspicious activities.

Task

Imagine that you are a security guard and you are monitoring two places. Someone informed you that there are suspicious activities in one of the places, but you were not told which one. Which place will you attend to?

Submit

TurKit (Little et al., 2010); Boto (http://code.google.com/p/boto/)

Human Computation Algorithms

human-driven operation

```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

TurKit (Little et al., 2010); Boto (http://code.google.com/p/boto/)

Human Computation Algorithms

human-driven operation

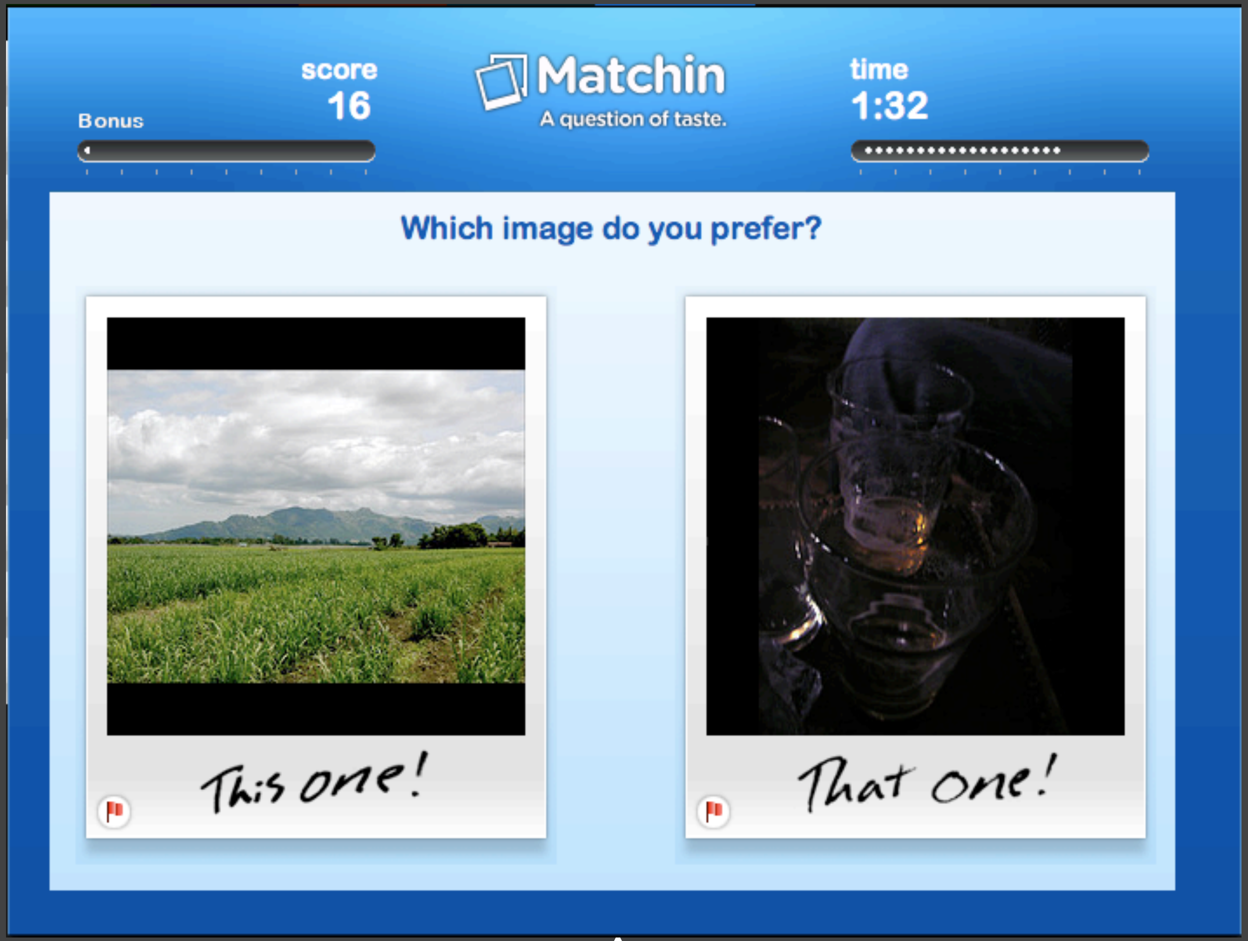
```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

Human Computation Algorithms

human-driven operation

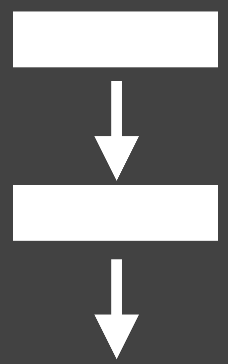
```
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

Games with a Purpose



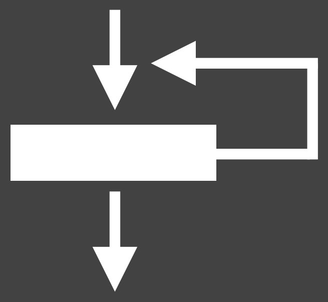
Human Computation Algorithms

human-driven controls



sequence

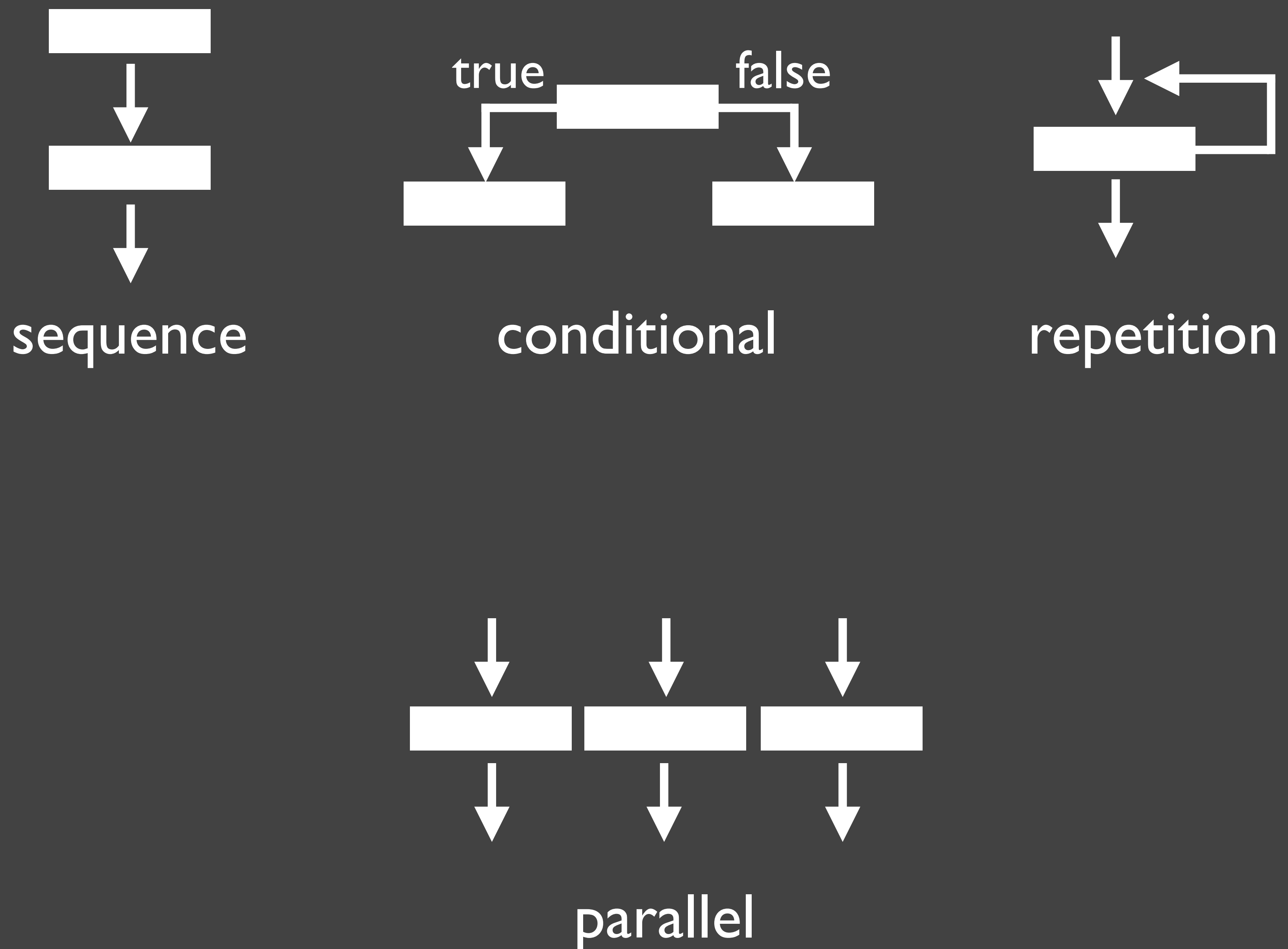
conditional



repetition

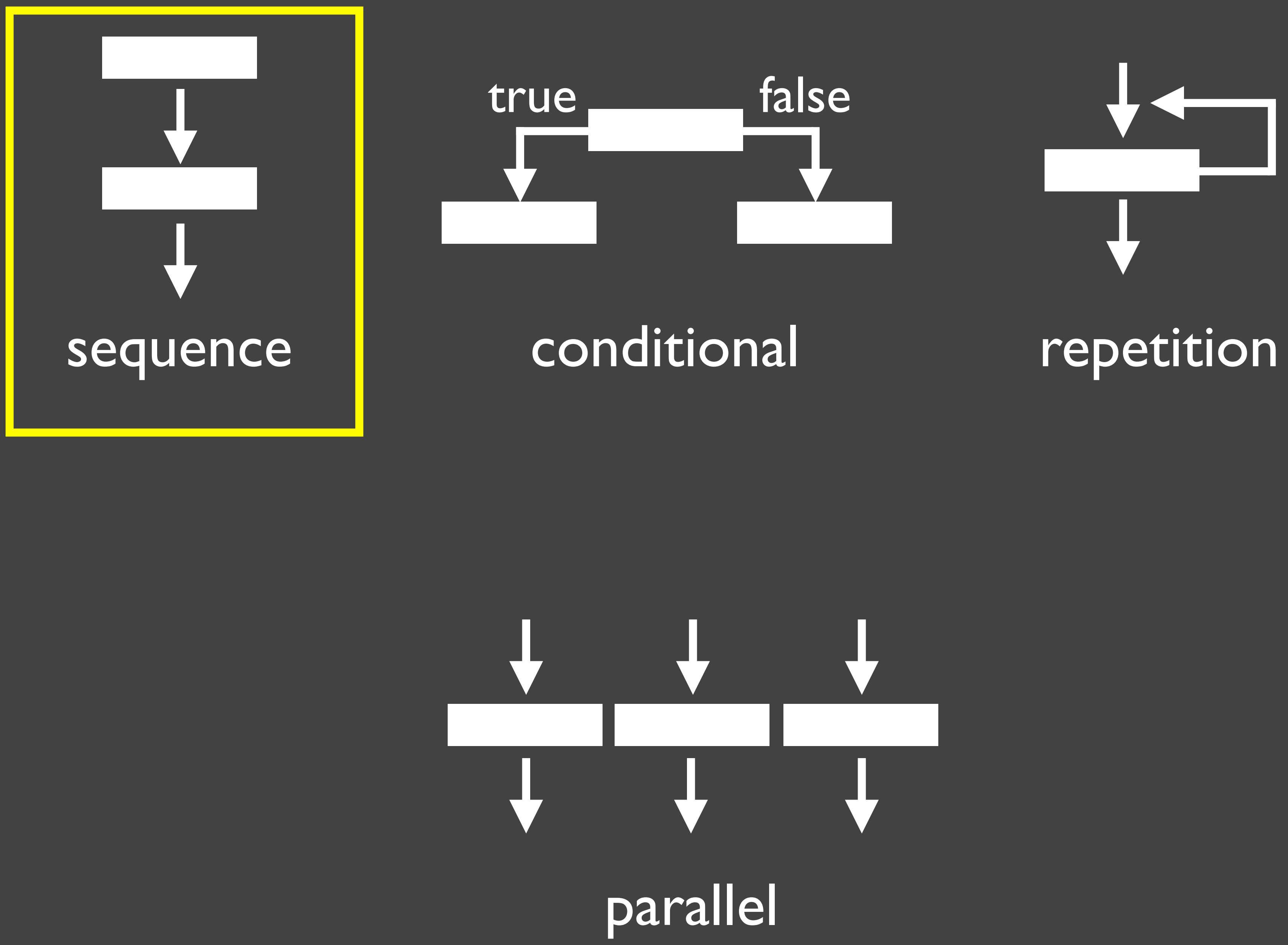
Human Computation Algorithms

human-driven controls



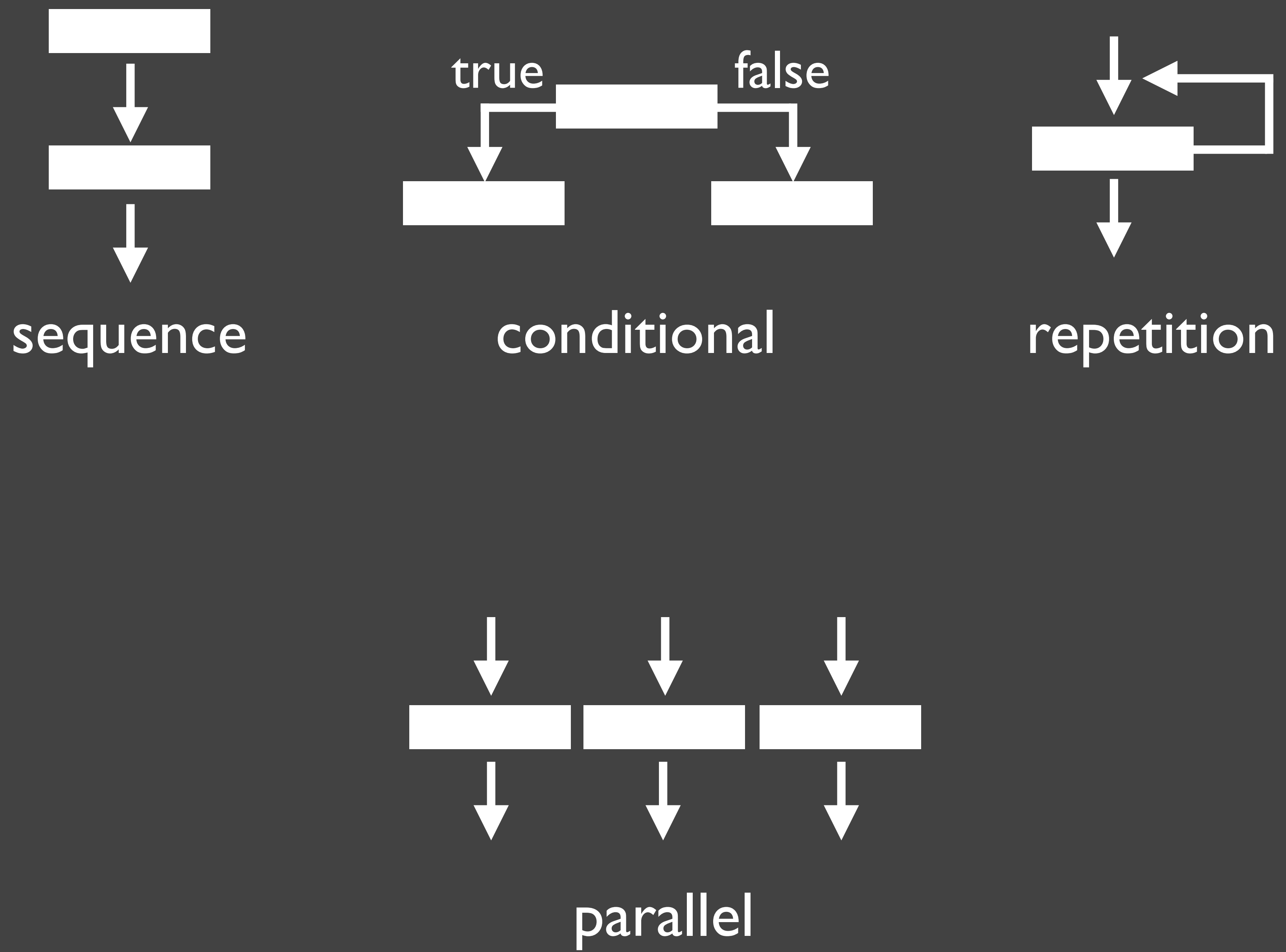
Human Computation Algorithms

human-driven controls



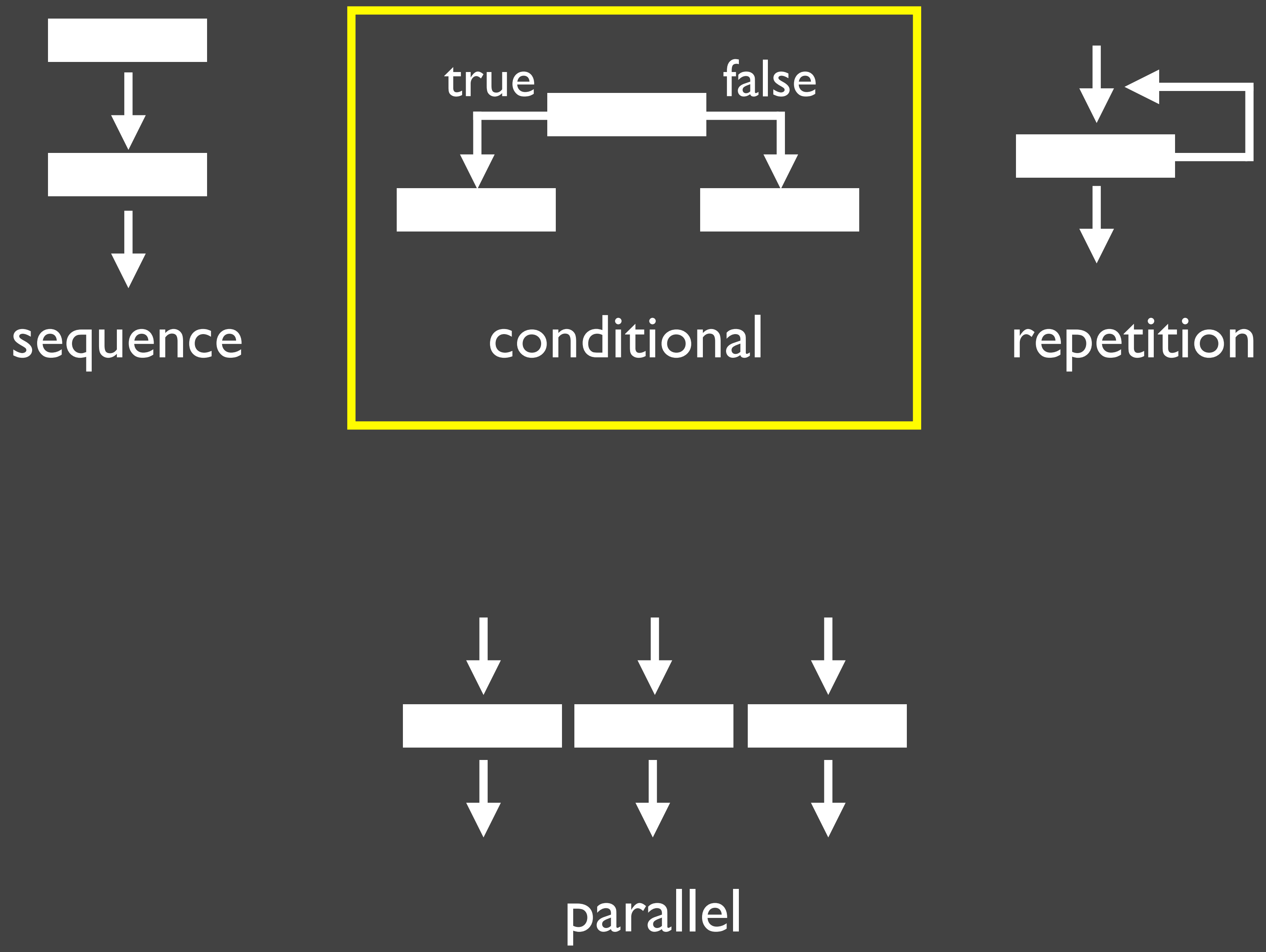
Human Computation Algorithms

human-driven controls



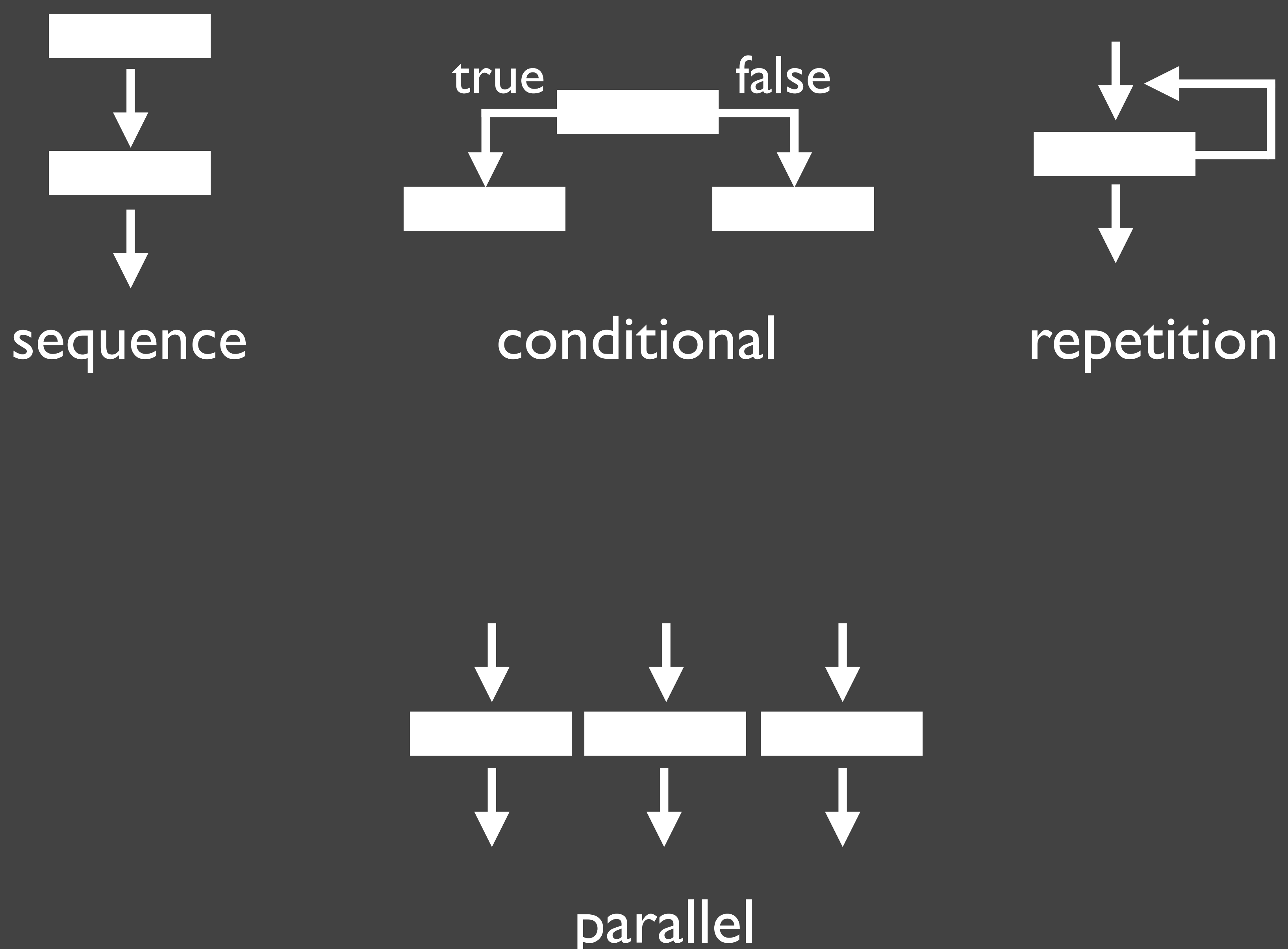
Human Computation Algorithms

human-driven controls



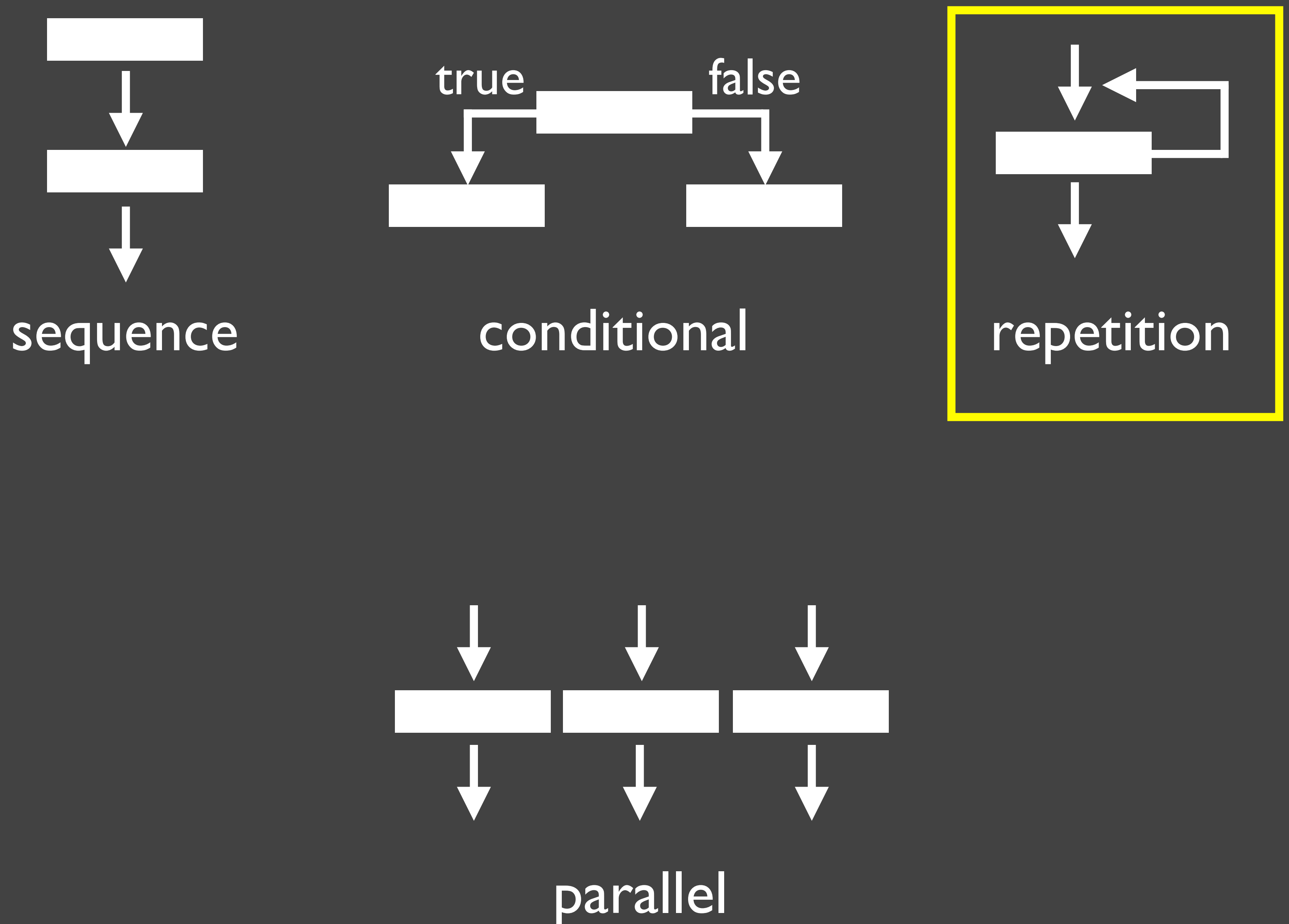
Human Computation Algorithms

human-driven controls



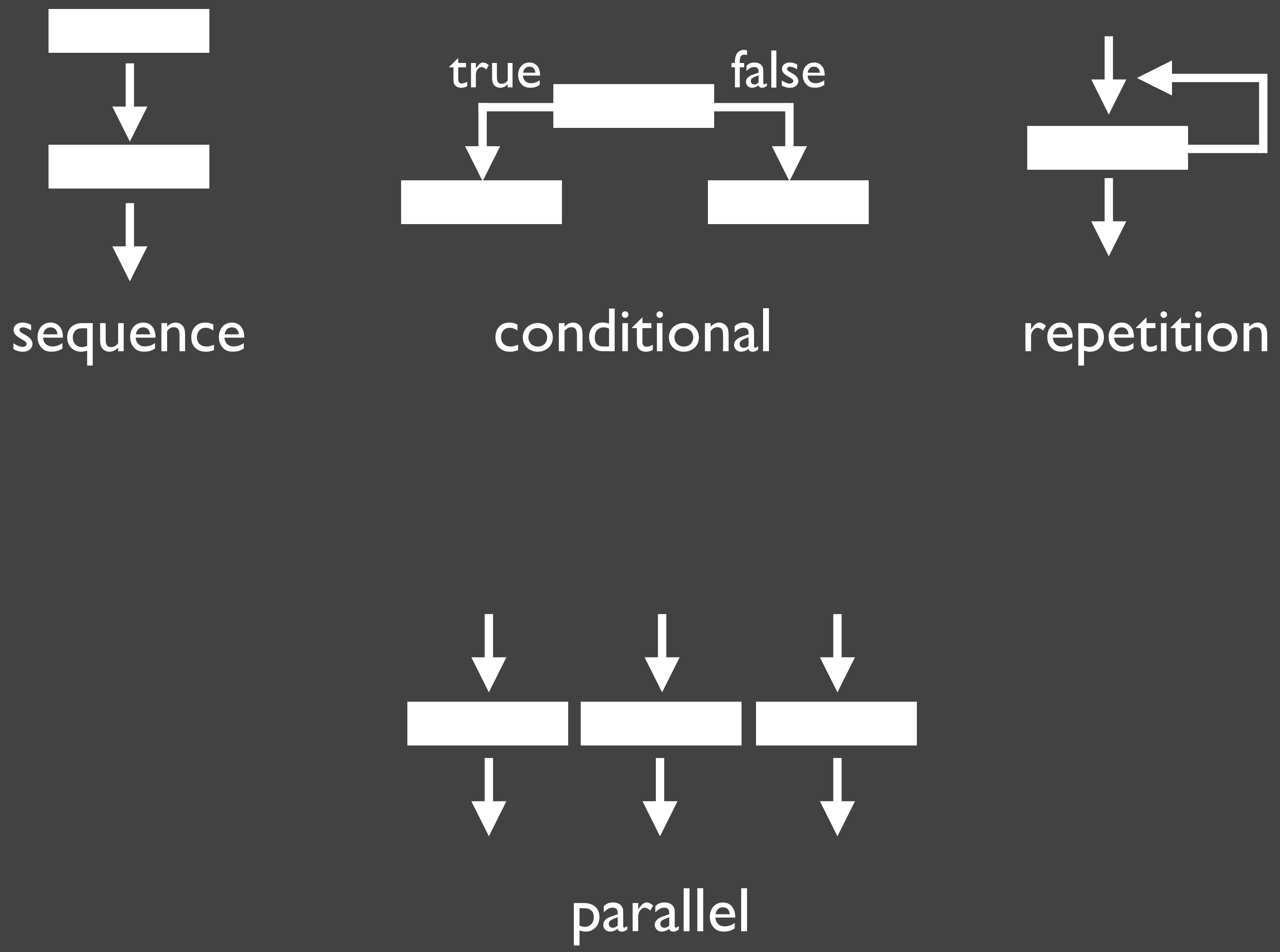
Human Computation Algorithms

human-driven controls



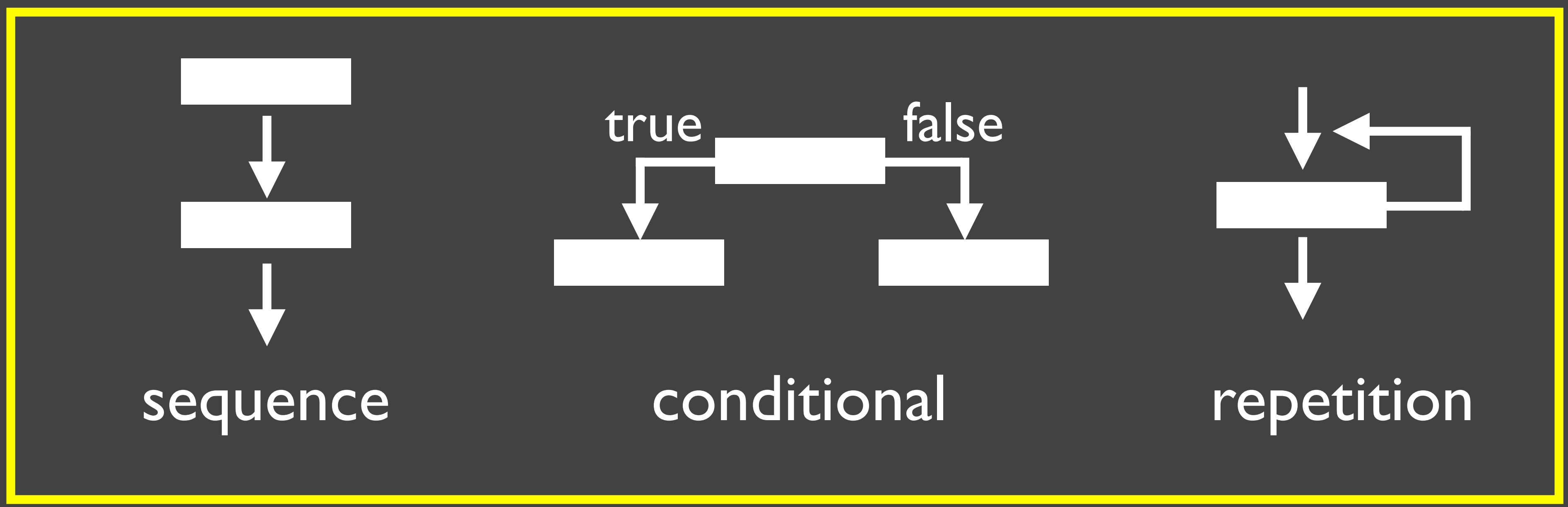
Human Computation Algorithms

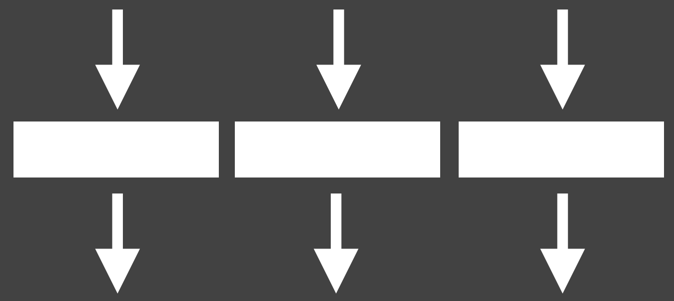
human-driven controls



Human Computation Algorithms

human-driven controls

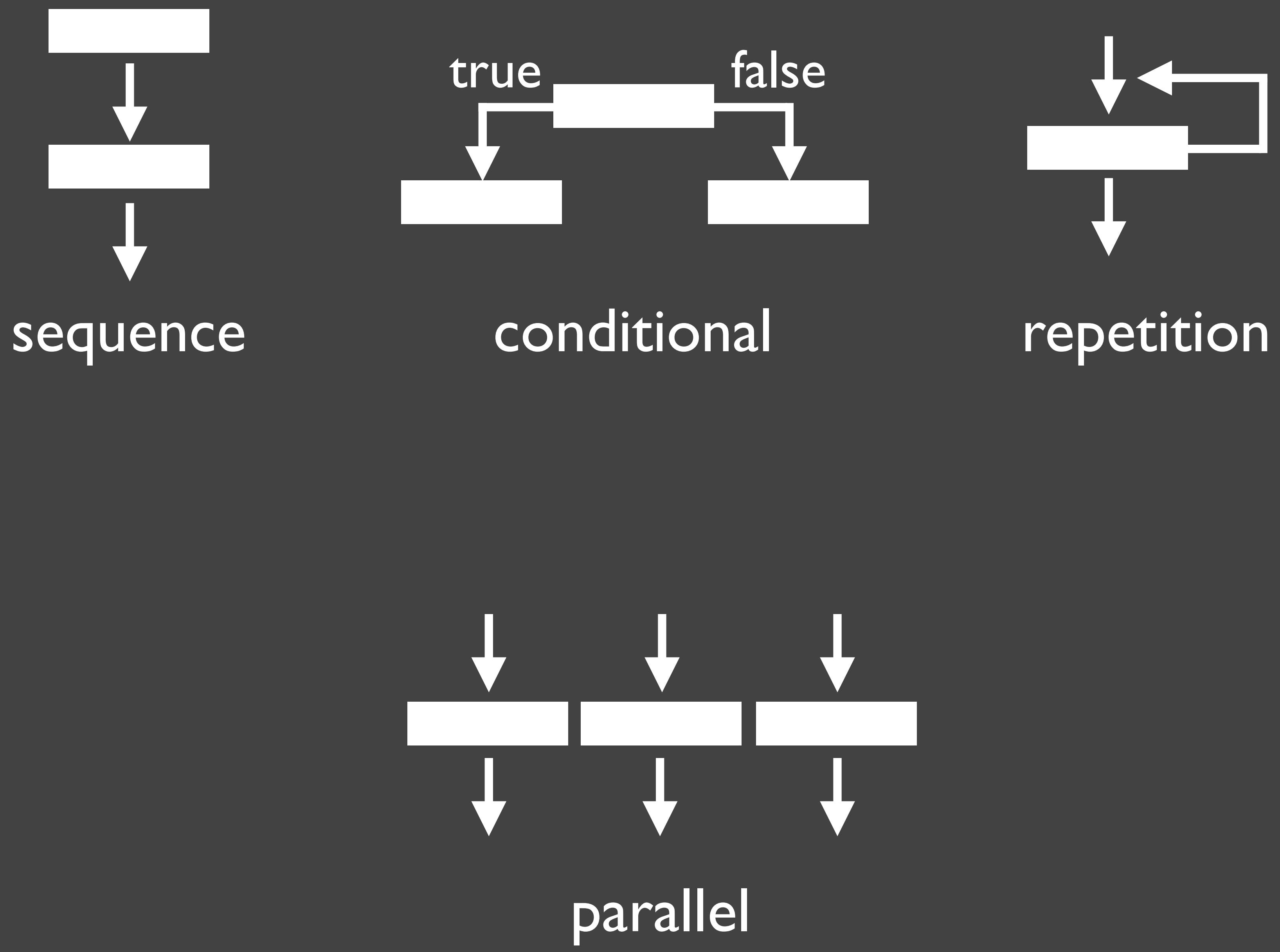




parallel

Human Computation Algorithms

human-driven controls



Human Computation Algorithms

human-driven program synthesis

Human Computation Algorithms

human-driven program synthesis

plan a wedding

Human Computation Algorithms

human-driven program synthesis

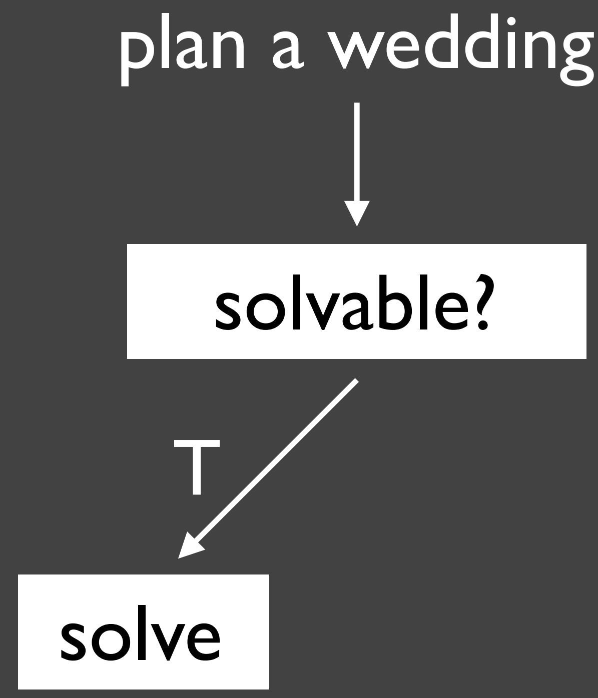
plan a wedding

↓

solvable?

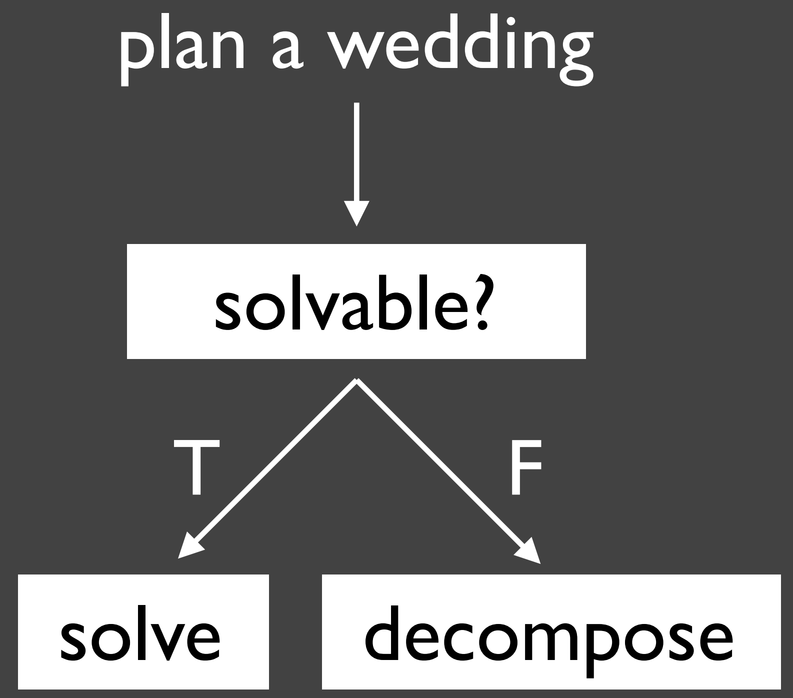
Human Computation Algorithms

human-driven program synthesis



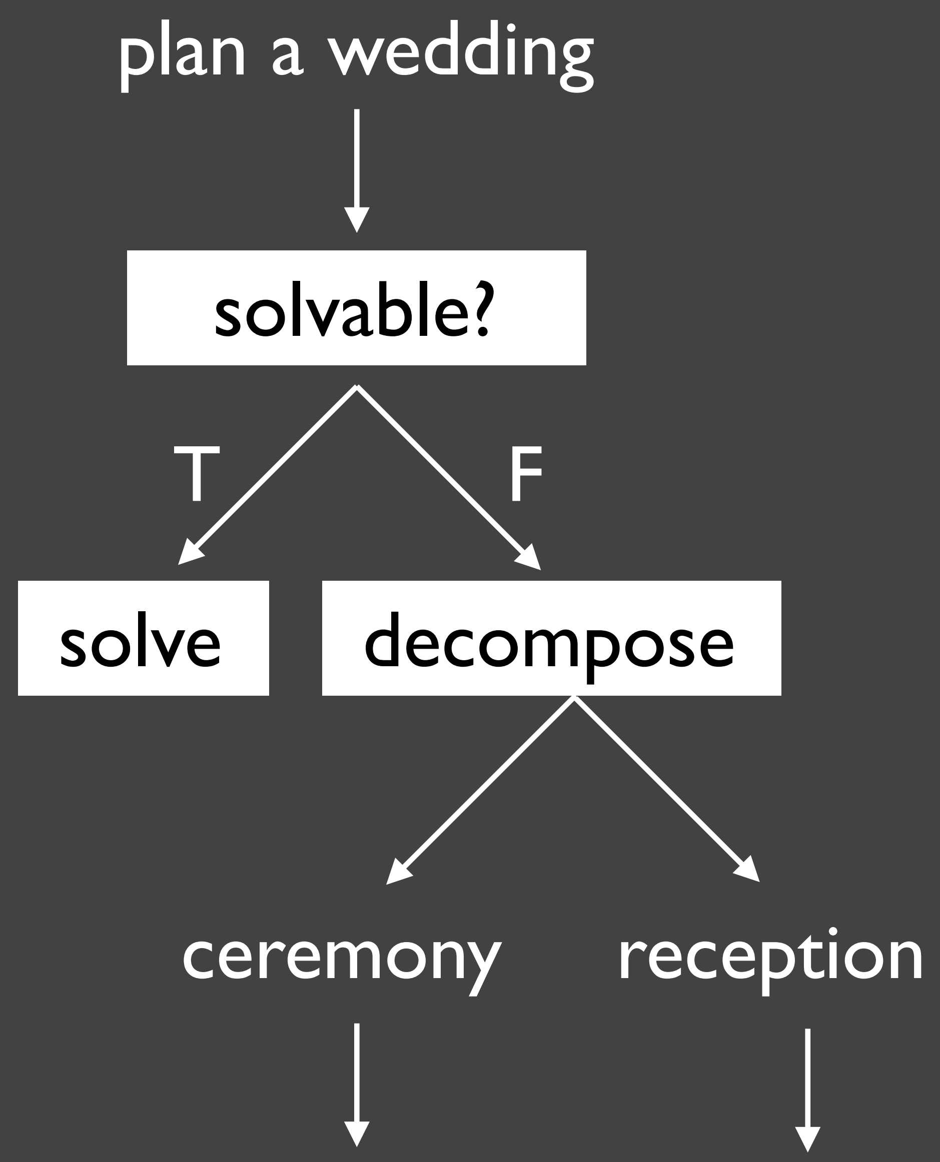
Human Computation Algorithms

human-driven program synthesis



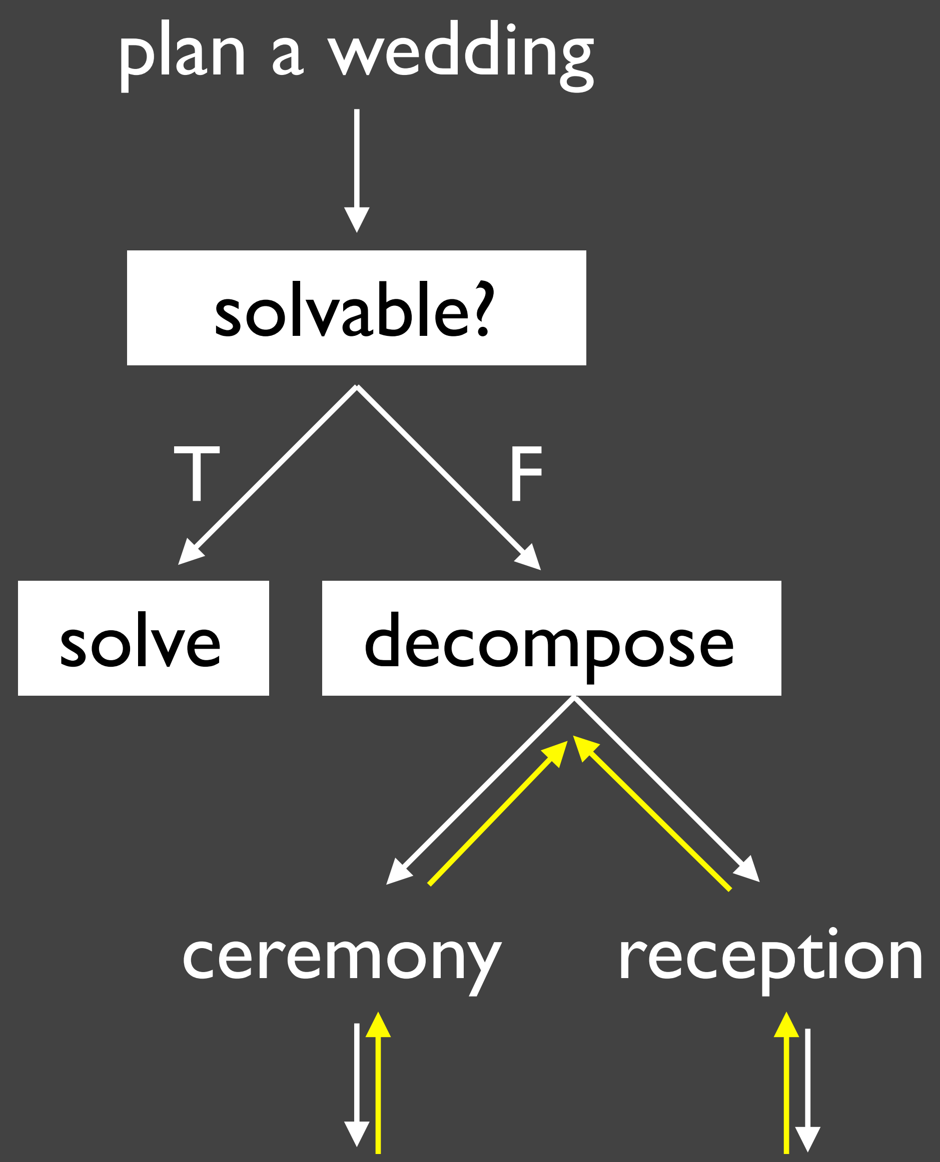
Human Computation Algorithms

human-driven program synthesis



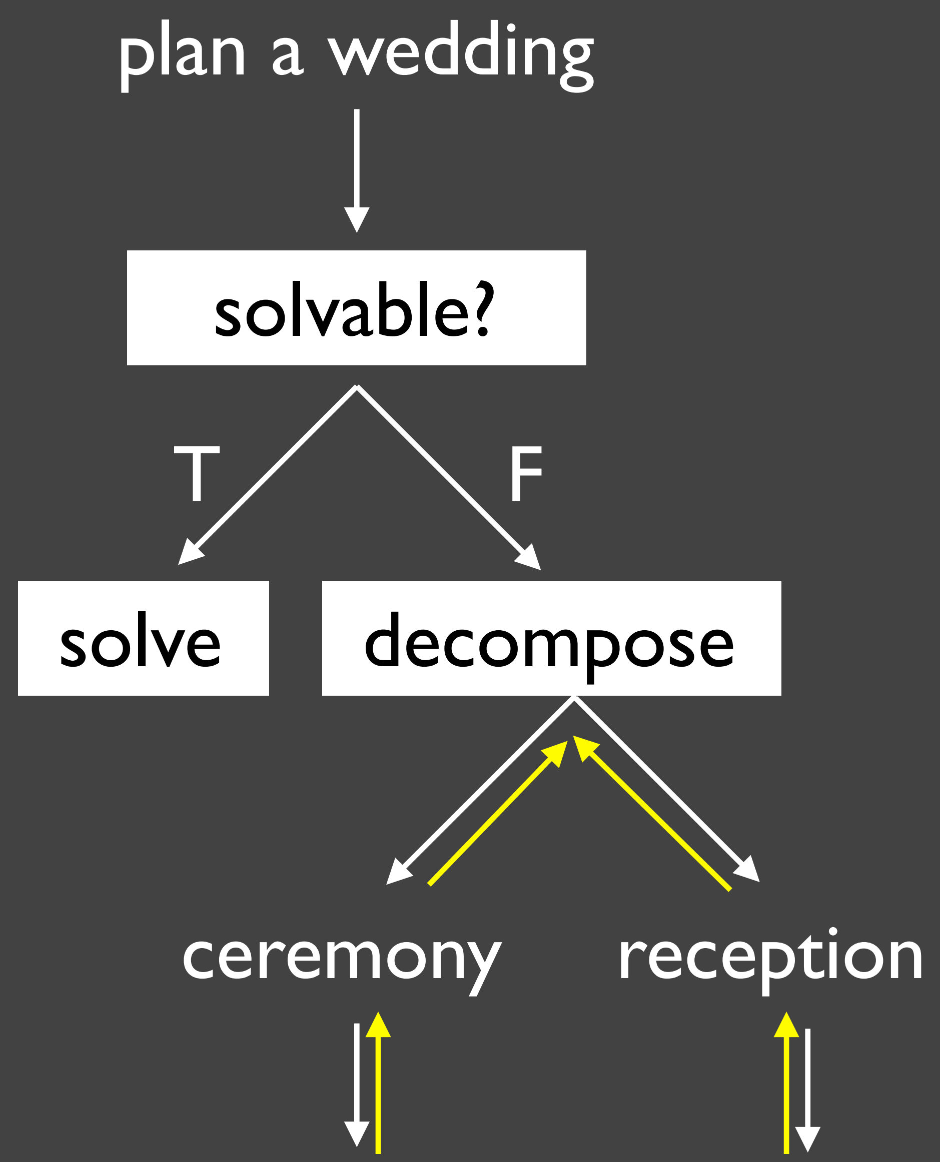
Human Computation Algorithms

human-driven program synthesis



Human Computation Algorithms

human-driven program synthesis

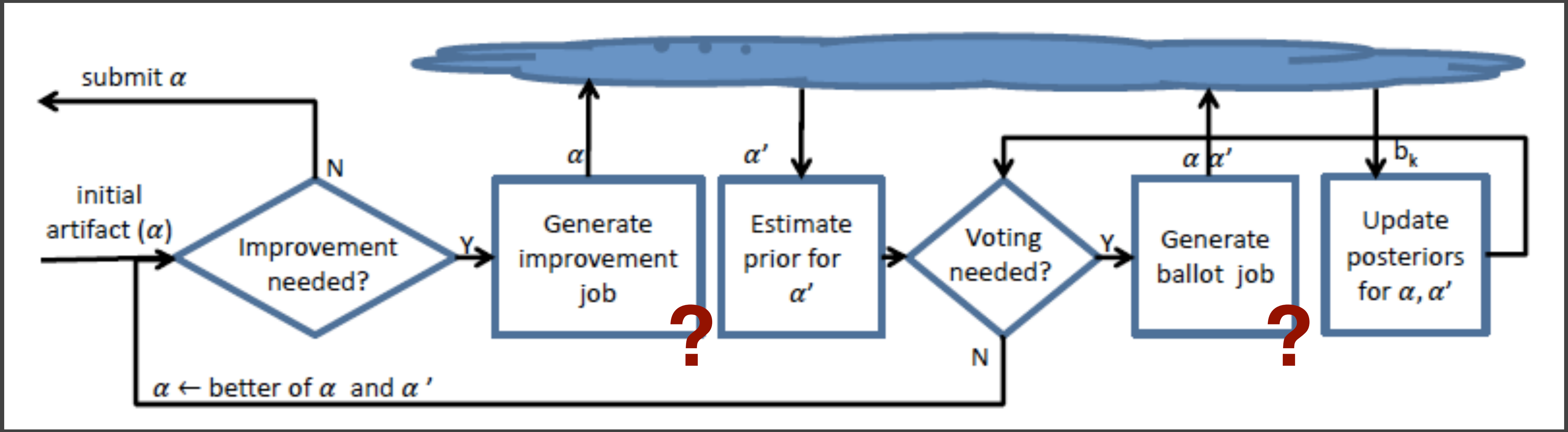


Turkomatic (Kulkarni et al., 2011)
CrowdForge (Kittur et al., 2011)

Human Computation Algorithms

automated design

Another improvement task? How many votes?



TurKontrol (Dai et al., 2010; 2011)

PROPERTIES

Is the algorithm correct?

Correctness

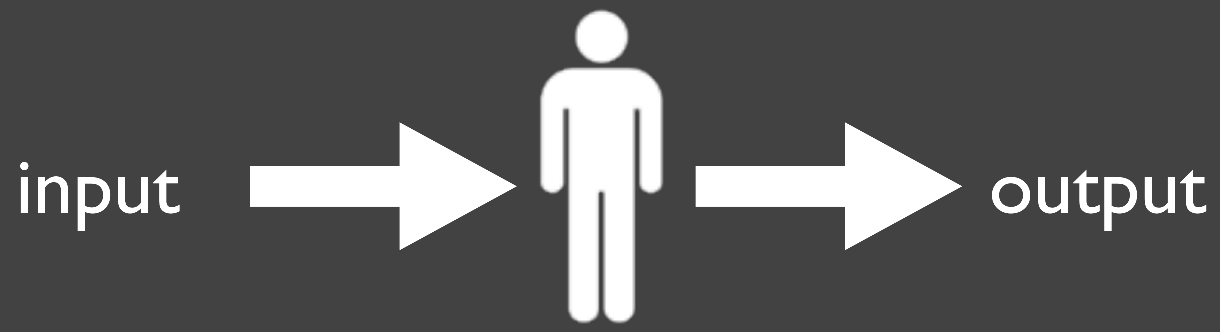
Theoretical Analysis

What does it mean for a human computation algorithm to be correct?

What guarantees can we give regarding the correctness of a human computation algorithm?

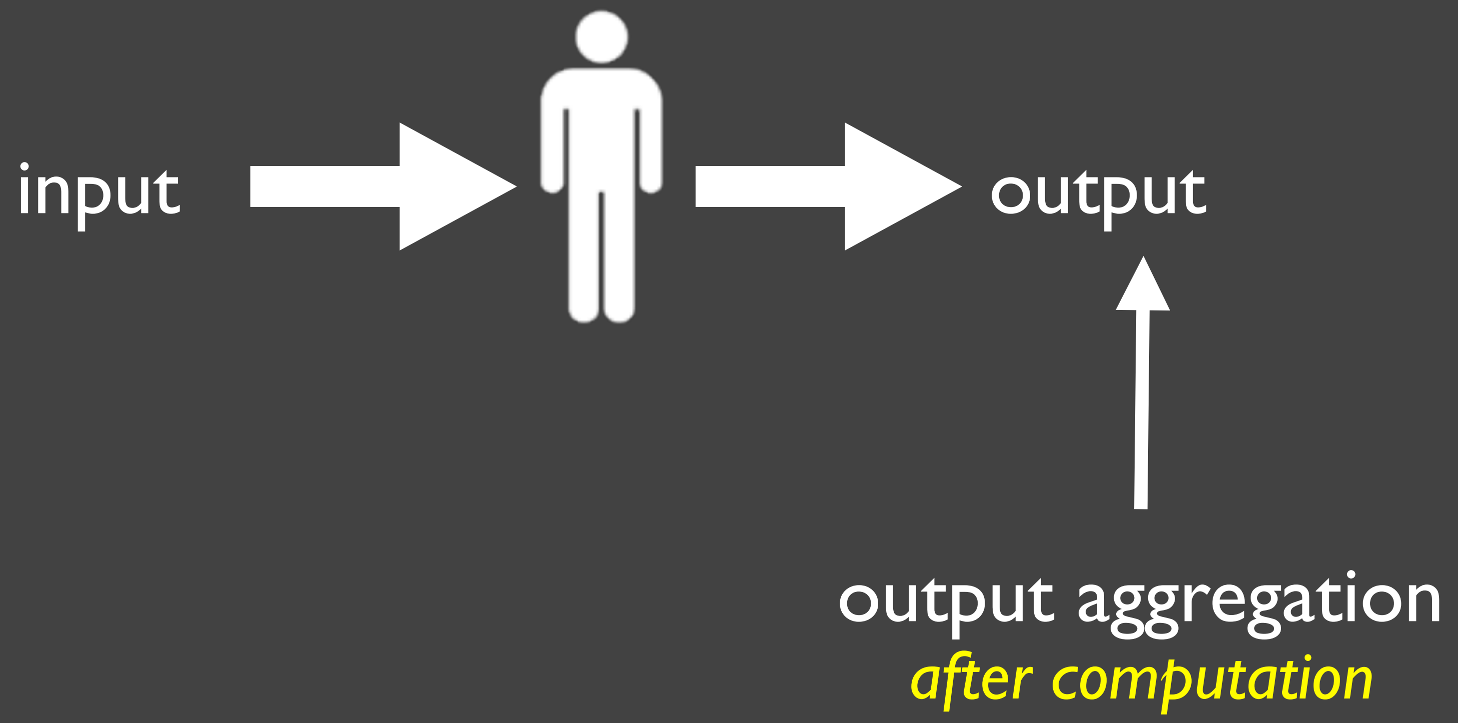
Correctness

Three Points of Intervention



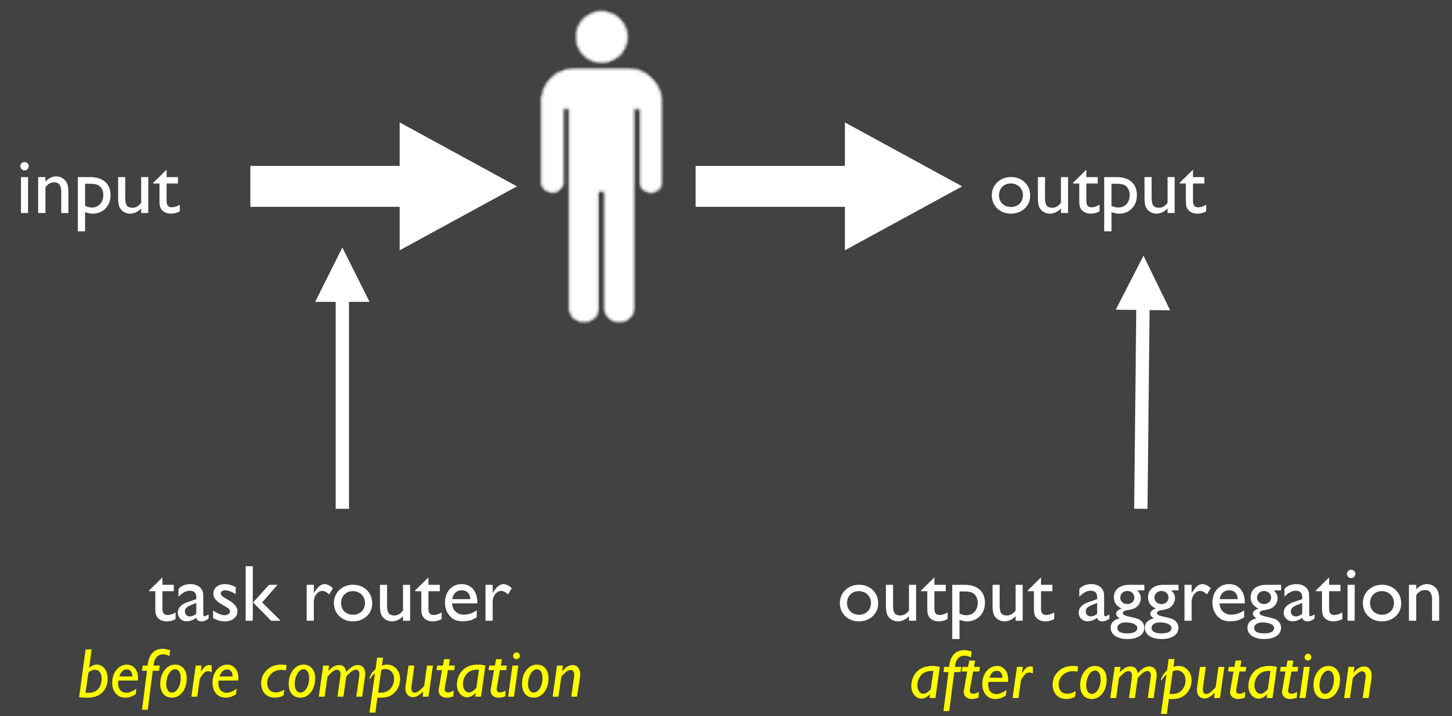
Correctness

Three Points of Intervention



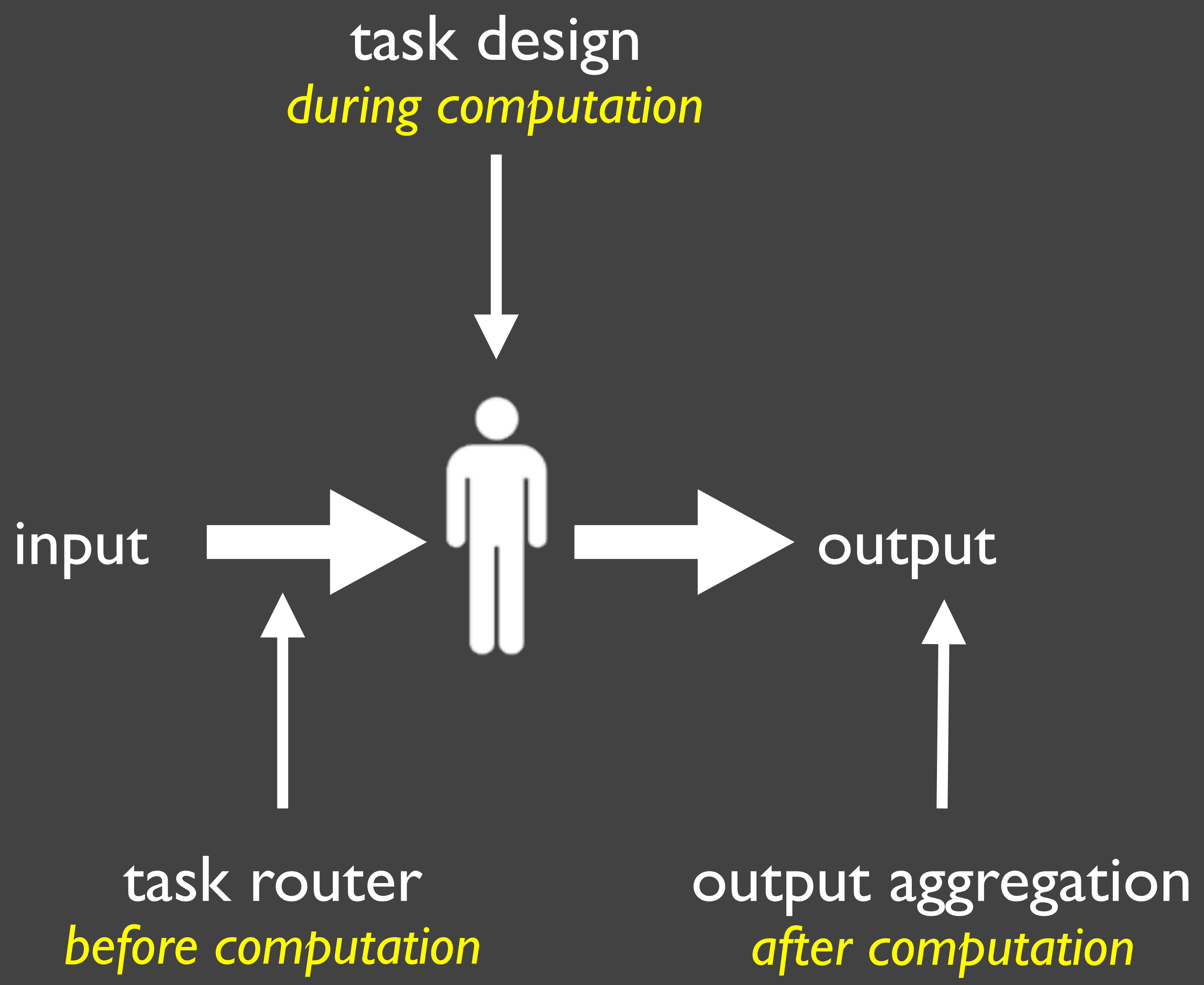
Correctness

Three Points of Intervention



Correctness

Three Points of Intervention



Is the algorithm efficient?

Efficiency

Three Measures

Time Complexity
How long does it take?

Query Complexity
How many queries to the human computers?

Cost Effectiveness
How much does it cost?

Efficiency

Time Complexity

Operation Complexity		Clock Time
How does the number of operations scale?	VS	How much time does it actually take?

Efficiency

The need for real-time

vizwiz & quikTurKit (Bigham et al., 2010)

Retainer Model (Bernstein et al., 2011)

Efficiency

Query Complexity

Efficiency

Query Complexity

1. Repeated Labeling

For each input object, how many human computers do we query?

Efficiency
Query Complexity

1. Repeated Labeling

For each input object, how many human computers do we query?

(Sheng et al., 2009; Kumar and Lease, 2011)

Efficiency

Query Complexity

Efficiency

Query Complexity

2. Active Learning

Which input should we process? What questions should we ask?

Active Learning
a short introduction

"The learner can select the data from which it learns."

(Settles, 2011)

Active Learning

a short introduction

"The learner can select the data from which it learns."
(Settles, 2011)

a single perfect oracle

label / feature / feature value

Active Learning

a short introduction

"The learner can select the data from which it learns."

(Settles, 2011)

a single pe~~r~~fect oracle

label / feature / feature value

Active Learning

a short introduction

"The learner can select the data from which it learns."

(Settles, 2011)

multiple imperfect oracles

a single perfect oracle

label / feature / feature value

Active Learning
a short introduction

"The learner can select the data from which it learns."
(Settles, 2011)

multiple imperfect oracles

a single perfect oracle X

label / feature / feature value X

Active Learning

a short introduction

"The learner can select the data from which it learns."

(Settles, 2011)

multiple imperfect oracles

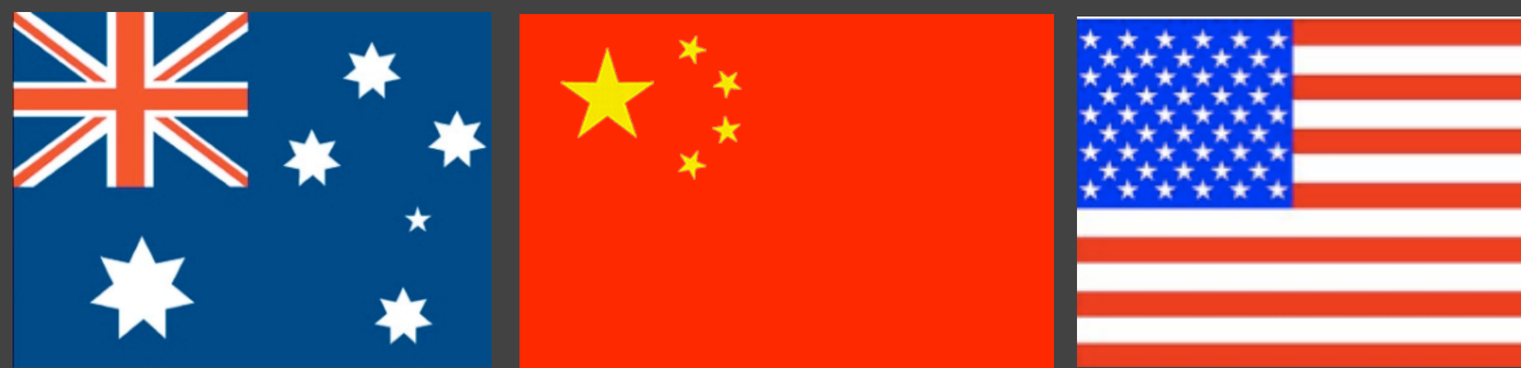
a single perfect oracle

richer, different kinds of questions

label / feature / feature value

Active Learning

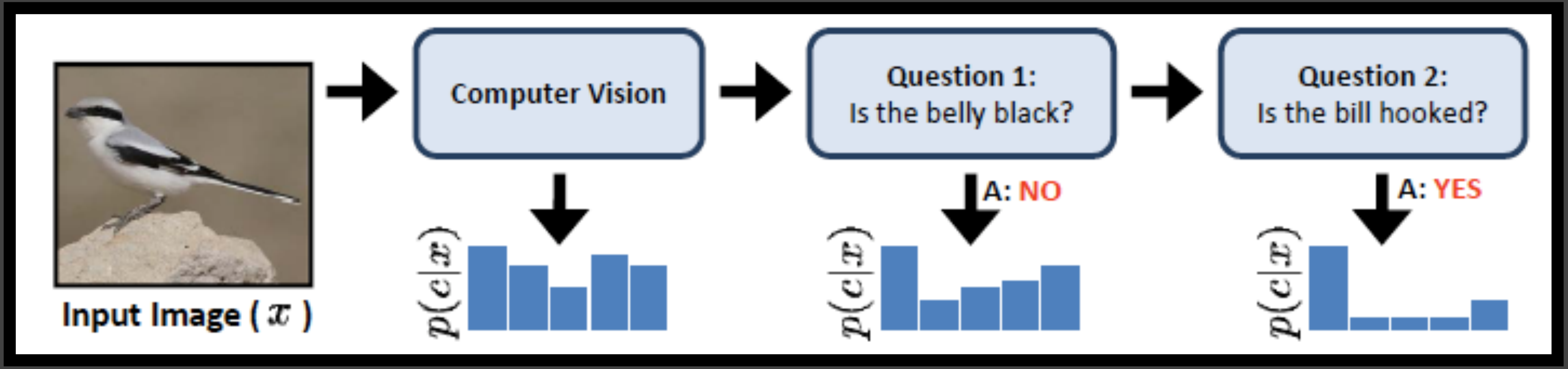
example # 1



(a) (b)

(Tamuz et al., 2011)

Active Learning
example # 2



(Branson et al., 2010)

Efficiency

Cost Effectiveness

Efficiency

Cost Effectiveness

$

How do we price each task?
Will the total cost be within budget?
What is the total cost in the worst case?
Can we minimize cost?
What is the cost-benefit tradeoff?

RECAP

TAKE-HOME

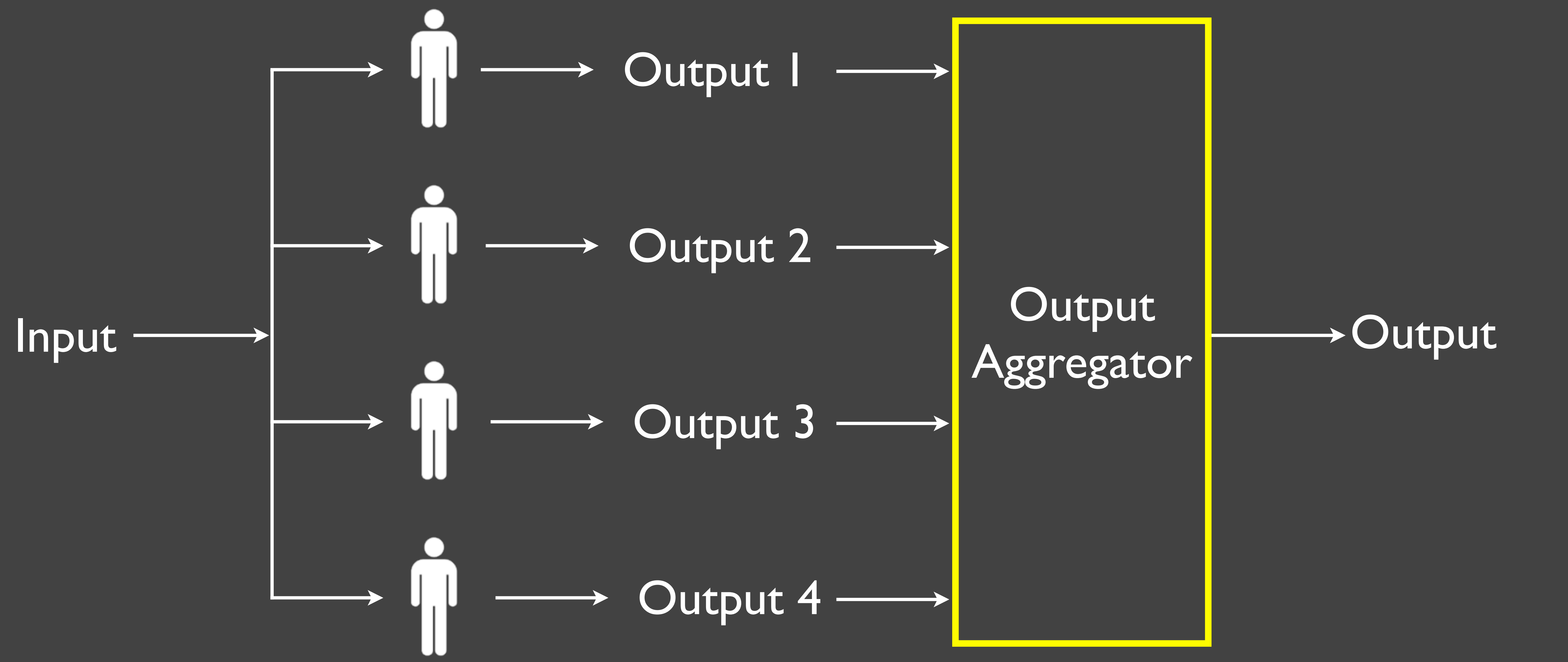
“human computation algorithms ↔ automated algorithms”

V

OUTPUT AGGREGATION

Motivation • Simple Outputs • Complex Outputs

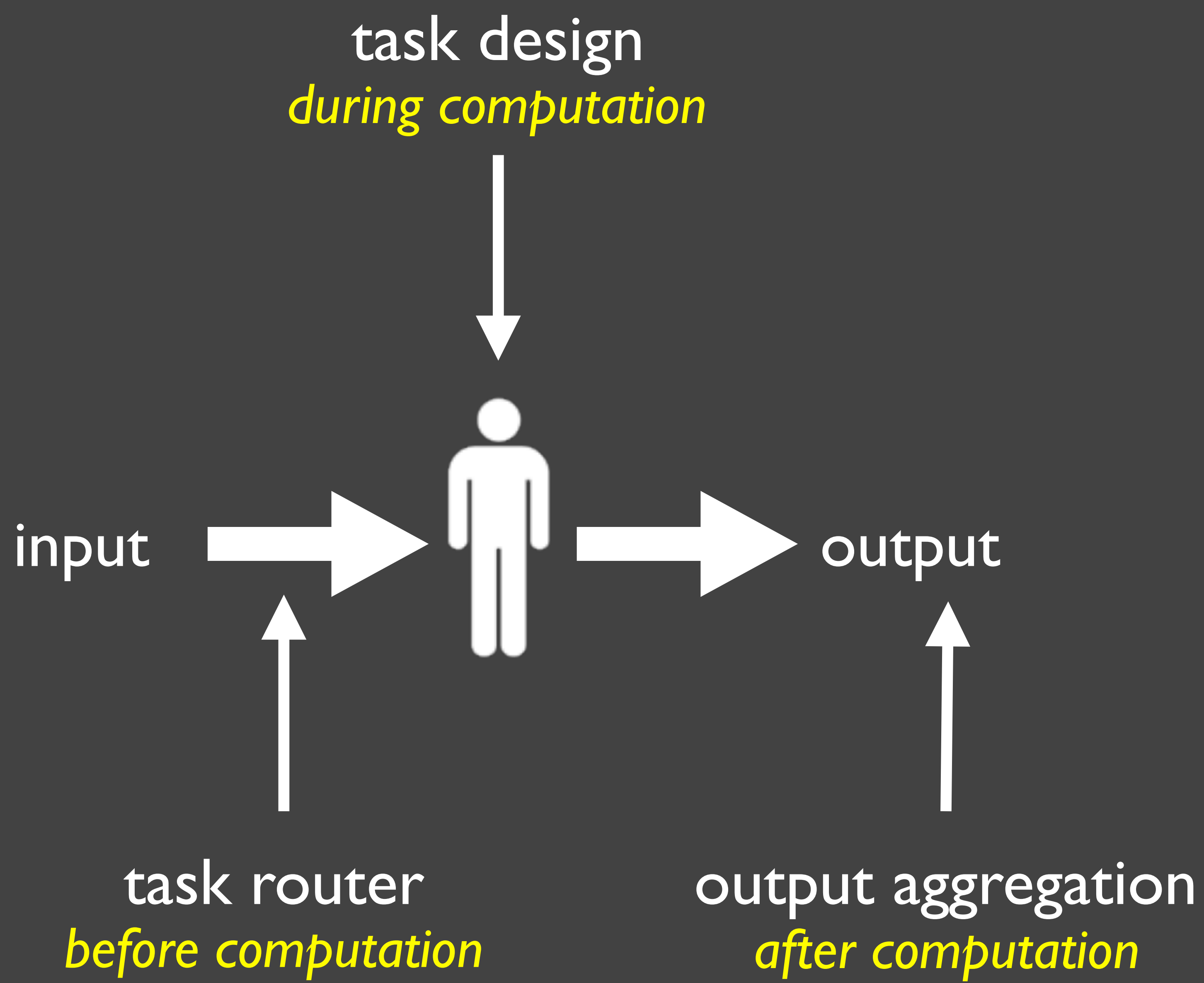
Output Aggregation
in a nutshell



Outputs can be aggregated by humans or automatically.

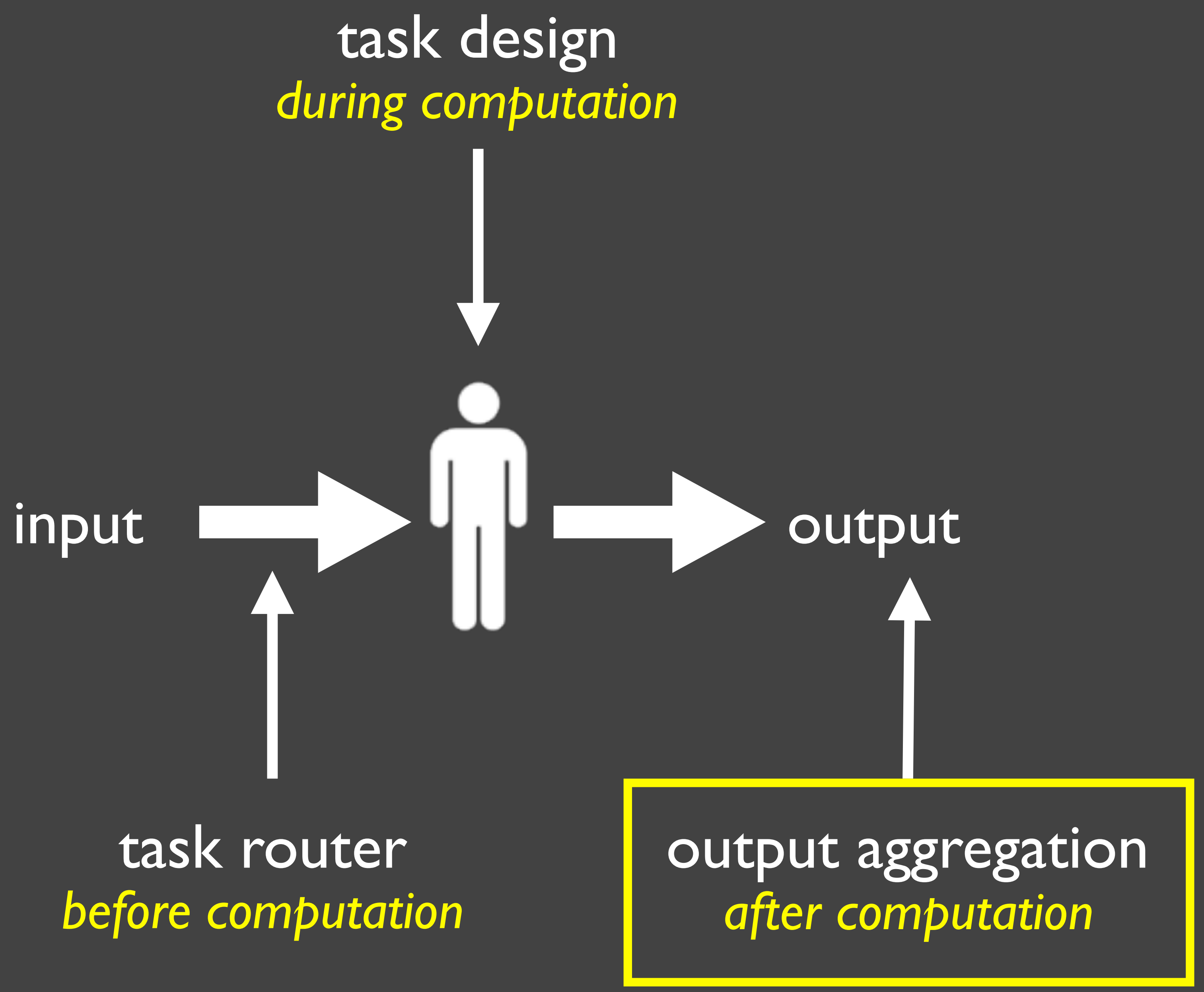
Correctness

Three Points of Intervention



Correctness

Three Points of Intervention



MOTIVATION

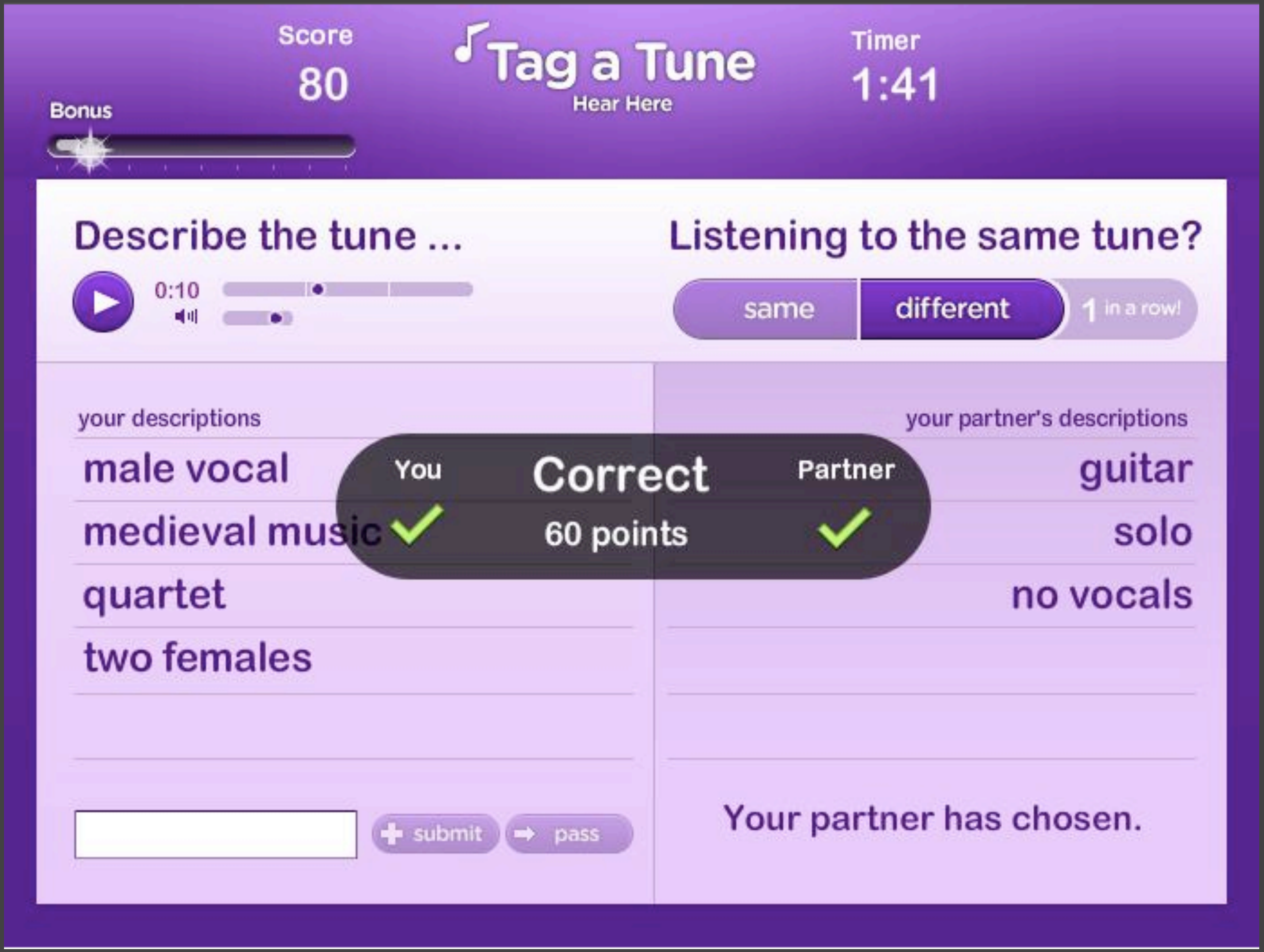
Outputs generated by human computers can be noisy.

Noise

is not only about inaccuracy

Noise

is not only about inaccuracy



(Law and von Ahn, 2009)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	**GRUITAR**	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	**ACCUSTIC**	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	**QUIET**	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	**SOFT**	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

Noise

example from TagATune

CLASSICAL	GRUITAR	FEMALE	RENNAISSANCE	STOMP
GUITAR	PRIMAL	VOCAL	SWING	SKIPPY
PIANO	ACCUSTIC	QUIET	SCI-FI	FOREIGN
VIOLIN	ACTIVE	SITAR	HIPPIE	CHRISTMASSY
ROCK	MEOW	CLASSIC	LULLABY	CLAPPY
SLOW	OHOHOH	SOFT	ANGELIC	CLOUDY
STRINGS	GRADUAL	CELLO	DOWNBEAT	SEASIDE
TECHNO	CLIMATIC	WOMAN	RELAXATION	MAMBO
OPERA	PENSIVE	MALE	GLOOMY	MANDOLIN
DRUMS	HOUSY	SINGING	ROYAL	FOLK
SAME	INSTRUMENTAL	VOCALS	RYTHMIC	NO VIOLINS
FLUTE	CALMISH	SOLO	MUFFLED	MELODY
FAST	FEMALE OPERA	LOUD	RAGTIME	HARMONICA
DIFF	VARIED	CHOIR	TUDOR	ITALIAN
ELECTRONIC	HEALING	VIOLINS	FANTASY	DRAMATIC
AMBIENT	WAVEY	HARP	HISPANIC	BLUEGRASS
BEAT	DRIPPING	BEATS	BEATLES	GENTLE
HARPSICORD	HEBREW	NOT ROCK	SYNCOPATED	SPACESHIP DESCENDING
SYNTH	ANIMALS	WIERD	MID-TEMPO	COOKIE MONSTER VOCAL
INDIAN	REEDS	DANCE	RATTLE	VAMPIRES AT A DINNER PARTY

(Law, Settles and Mitchell, 2010)

The “truth” exists, and
through redundancy we can find it.

Truth

objective versus cultural

Objective Truth	Cultural Truth

Truth

objective versus cultural

Objective Truth

a definitive answer exists beyond human judgments, but hard to reach.

e.g., cancer or not
number of volcanos on Venus
location or time of a photo

Cultural Truth

Truth
objective versus cultural

Objective Truth

a definitive answer exists beyond human judgments, but hard to reach.

e.g., cancer or not
number of volcanos on Venus
location or time of a photo

Cultural Truth

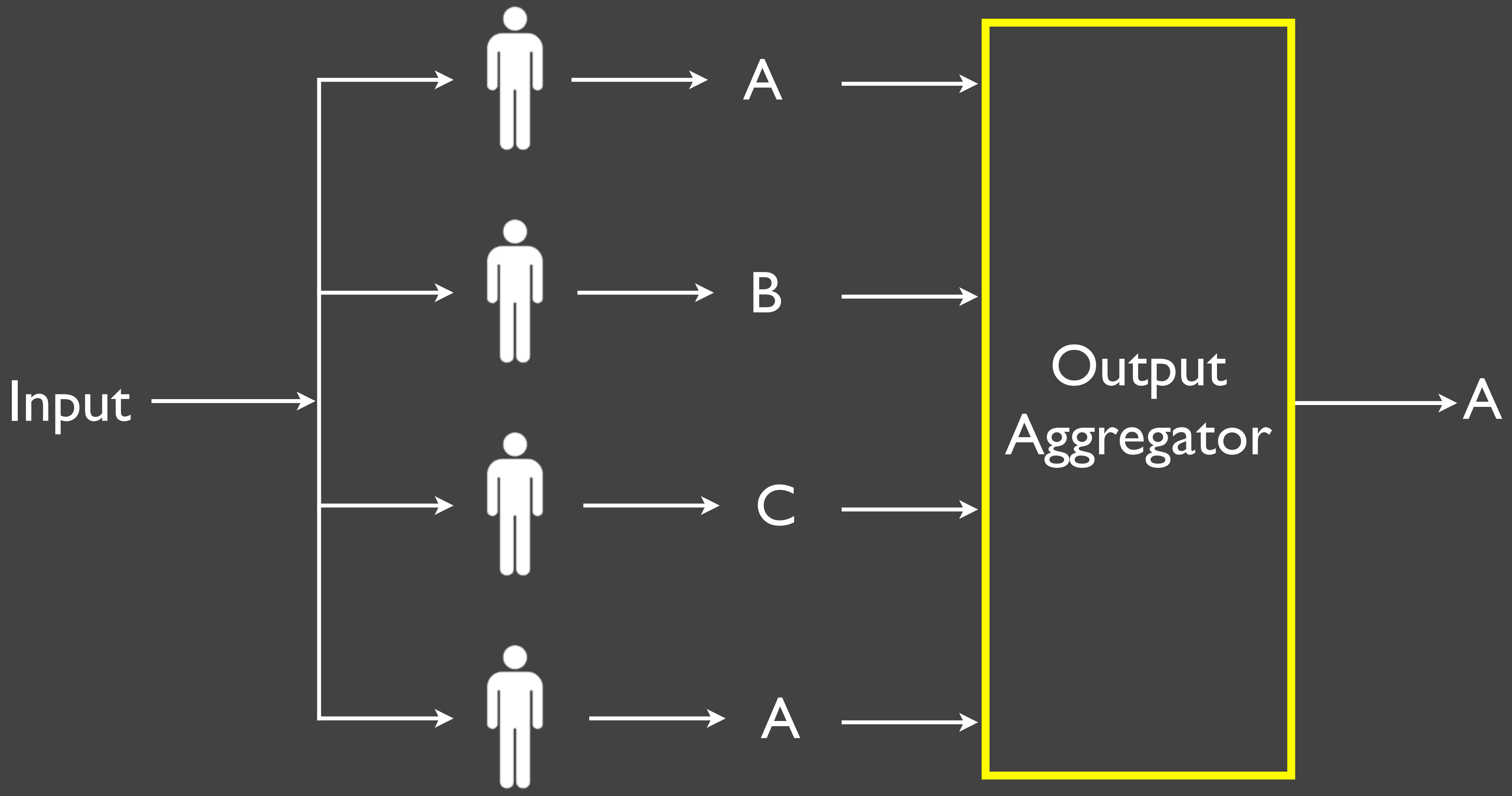
shared beliefs of a group of people, often involving perceptual judgments.

e.g., is this music calm?
is this image pornographic?
is this disease contagious?

SIMPLE OUTPUTS

Output Aggregation

classification



Statistical Measures of Agreement

(Artstein and Poesio, 2008)

Statistical Measures of Agreement

(Artstein and Poesio, 2008)

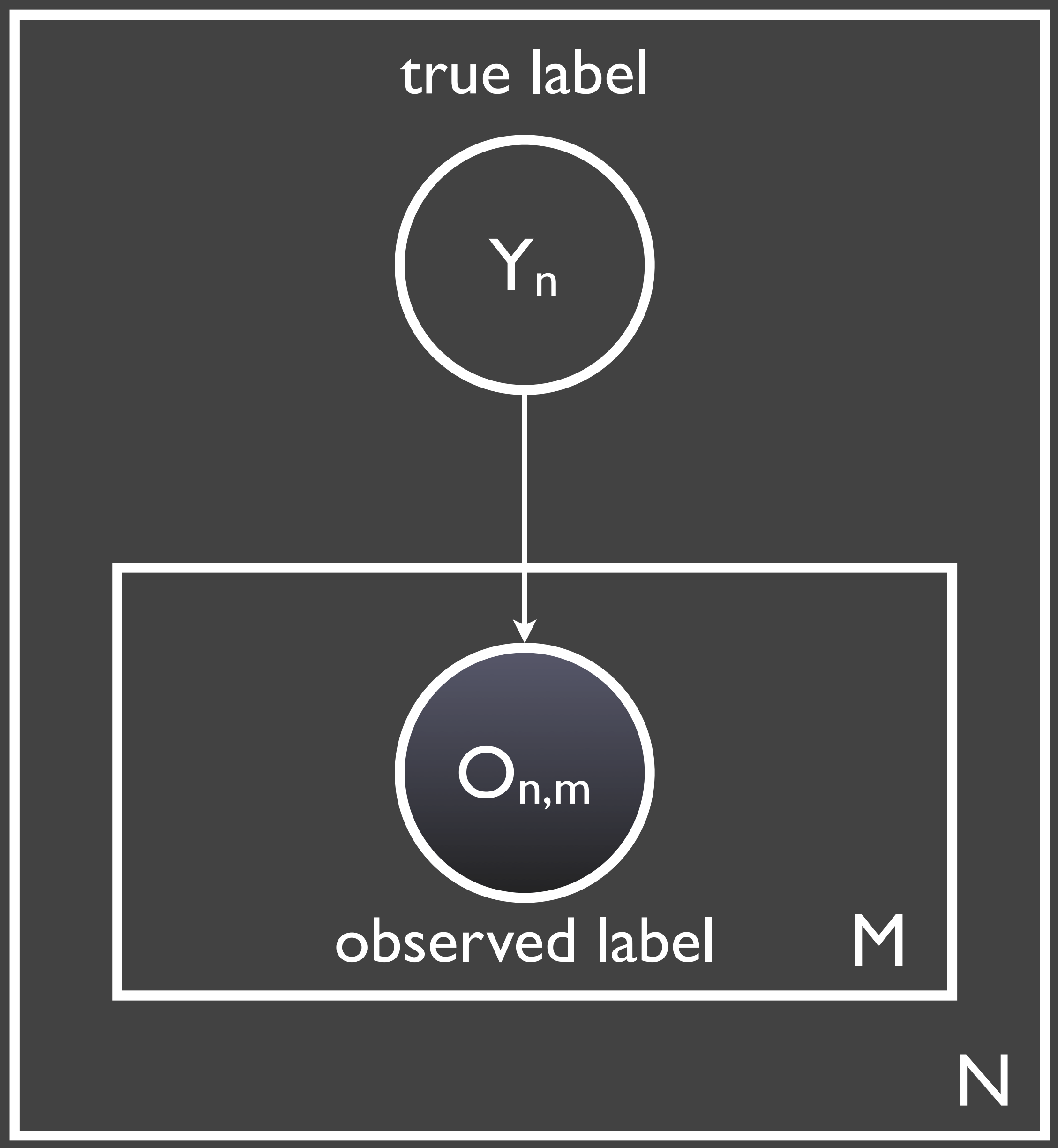
The simplest way to aggregate is majority vote.

Statistical Measures of Agreement
(Artstein and Poesio, 2008)

The simplest way to aggregate is majority vote.

But how much agreement is really there?

Majority Vote
as a graphical model



N classification questions, M workers

Hidden Factors

that influence the annotation process

Hidden Factors
that influence the annotation process

Task Characteristics
- Quality (e.g., blurry pictures)
- Difficulty (e.g., transcription of non-native speech)

Hidden Factors

that influence the annotation process

Task Characteristics

- Quality (e.g., blurry pictures)
- Difficulty (e.g., transcription of non-native speech)

Worker Characteristics

- Expertise (e.g., bird identification)
- Bias (e.g., mother vs college students)
- Physical Conditions (e.g., fatigue)

Latent Class Model

for classification

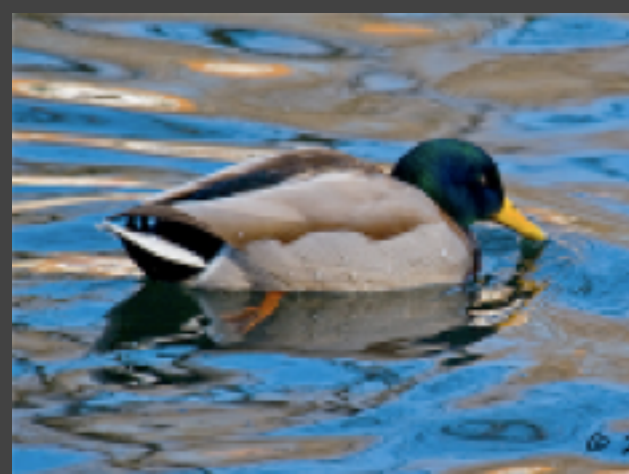
Latent Class Model
for classification

Dawid and Skeen, 1979
Uebersax et al., 1993
Carpenter, 2008
Whitehill et al., 2009

Ipeirotis et al., 2010
Raykar et al., 2010
Welinder and Perona, 2010
Ipeirotis et al., 2010

Latent Class Model

an example

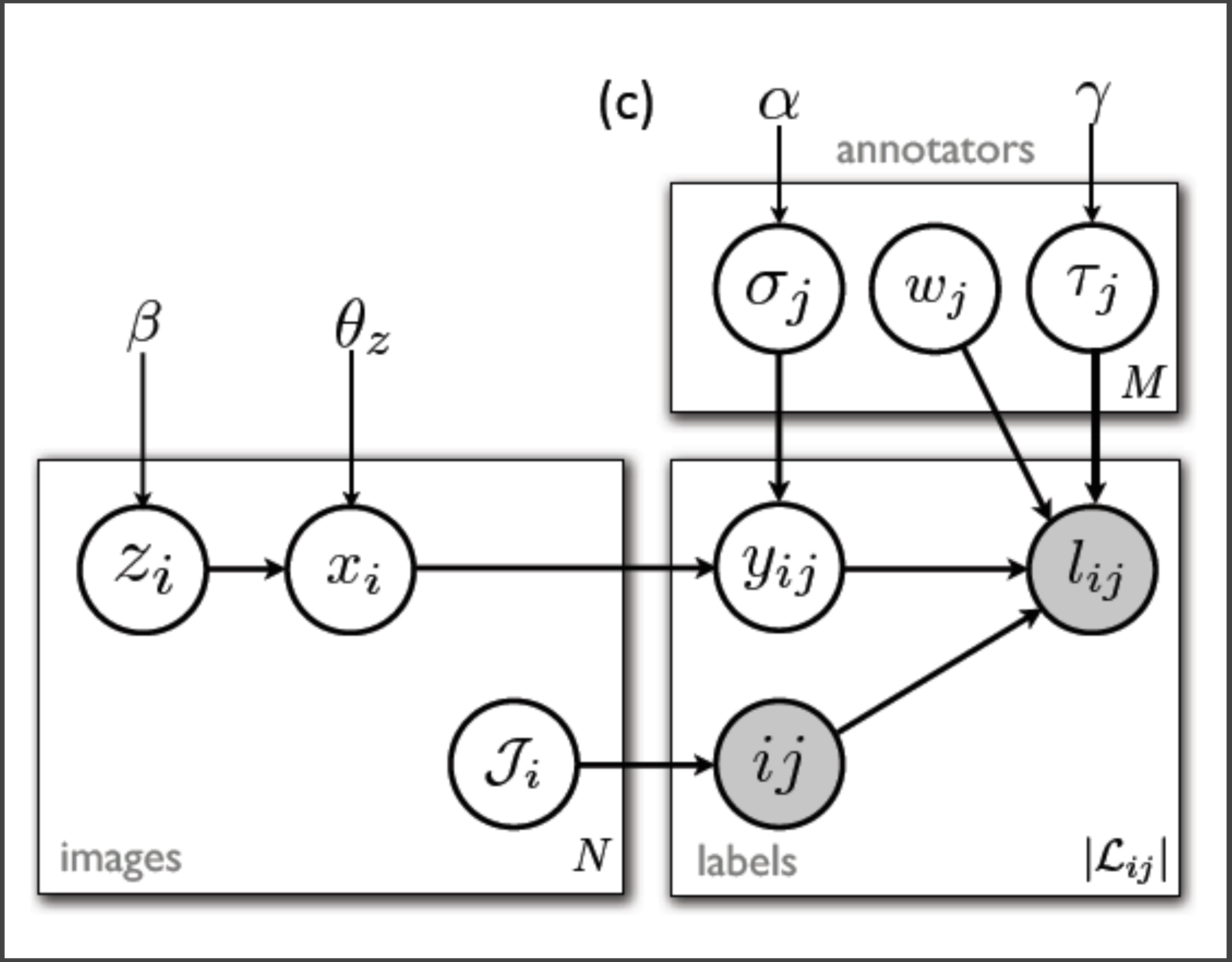


Grebe?

(Welinder et al., 2010)

Latent Class Model

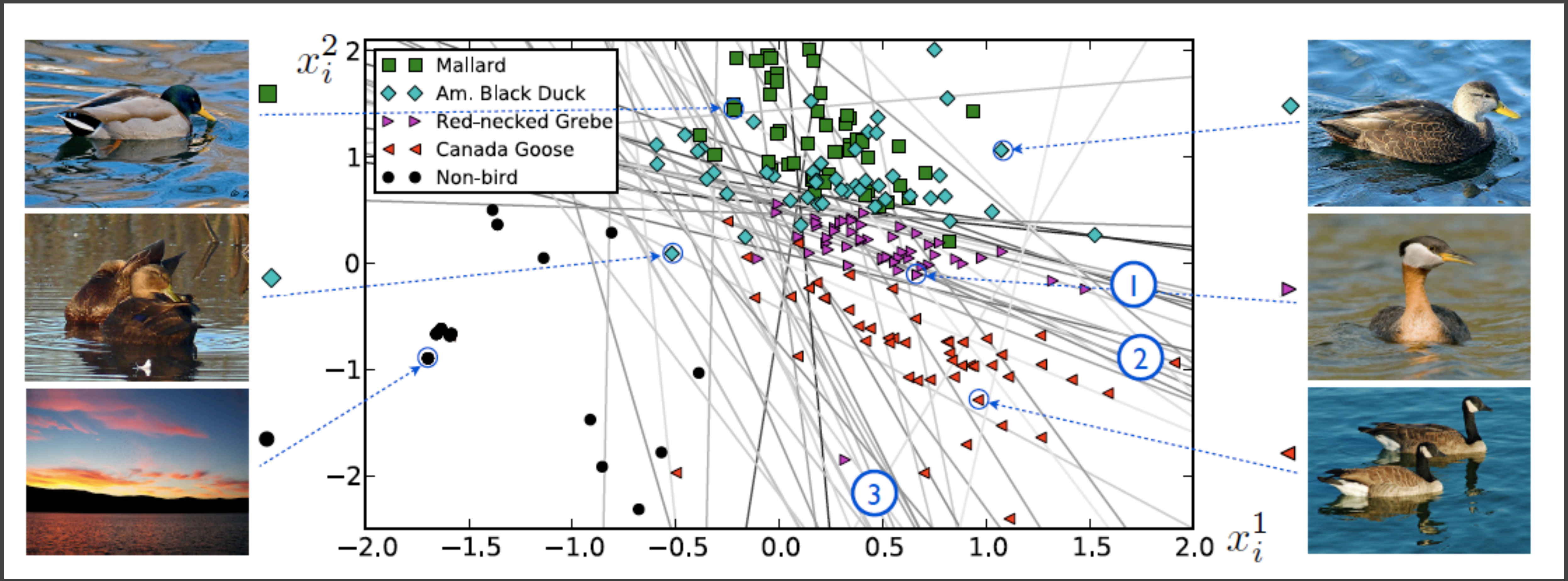
an example



(Welinder et al., 2010)

Latent Class Model

an example



(Welinder et al., 2010)

Cultural Consensus Theory

in Anthropology

	Respondent																							
	10	17	9	23	22	6	18	2	3	5	8	21	1	14	7	13	16	4	20	12	19	15	24	
Allergies	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	20**
Kidney pain	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0	19**
Gastritis	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	0	0	0	1	0	1	16*
Amoebas	1	0	1	1	1	1	1	1	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	15
Appendicitis	1	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	0	15
Hepatitis	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1	1	0	14
Mumps	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	0	0	13
Rubella	1	1	0	1	1	0	1	0	1	0	0	0	1	1	0	0	1	1	1	1	1	0	0	13
Measles	1	1	0	1	1	0	1	0	1	1	0	0	1	1	0	0	0	1	1	1	1	0	0	13
Smallpox	1	1	1	1	1	0	1	0	1	1	0	0	1	1	0	0	1	0	0	1	1	0	0	13
Cancer	1	1	1	0	0	1	0	1	0	1	1	1	0	0	1	0	0	0	1	0	0	1	1	12
Diabetes	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	1	0	1	1	0	12
Intestinal influenza	1	0	1	1	1	1	1	1	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	12
Tetanus	1	1	1	1	0	1	0	1	1	0	1	1	0	0	1	0	0	0	1	0	0	0	1	12
Chicken pox	1	1	0	1	1	0	1	0	1	0	0	0	1	1	0	0	1	1	0	1	1	0	0	12
Tonsilitis	0	0	1	1	0	0	1	1	1	0	1	1	0	1	0	1	0	1	0	0	1	0	0	11
Polio	0	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	11
Diarrhea	0	0	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	10
Typhoid fever	0	1	0	1	1	0	0	1	1	1	1	0	0	1	0	0	0	1	1	0	0	0	0	10
Diphtheria	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	1	0	1	7*
Arthritis	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	6**
Whooping cough	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	6**
Tuberculosis	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	6**
Malaria	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	5**
Colic	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	4**
Rheumatism	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	3**
Flu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1**
Errors	2	3	7	6	4	7	4	10	6	6	9	8	4	6	8	8	6	9	9	6	7	12	12	

from (Romney, 1999)

(Romney et al., 1986; Karabatsos and Batchelder, 2003; Weller, 2007)

Other Challenges

What if we cannot assume repeated labeling?

Other Challenges

What if we cannot assume repeated labeling?

“Learn a hypothesis to simulate the aggregate output, and prune away workers that don’t agree”

(Dekel and Shamir, 2009)

COMPLEX OUTPUTS

Complex Outputs

and challenges

ranking & clustering

structured outputs

beliefs

Challenge #1:

deciding how to decompose the problem

Ranking Aggregation

individual rankings ➠ full ranking

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)

Ranking Aggregation

individual rankings ➠ full ranking

paired comparison

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)

Ranking Aggregation

individual rankings ➠ full ranking

paired comparison

rating (scale 1-4)

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)

Ranking Aggregation

individual rankings ➠ full ranking

paired comparison

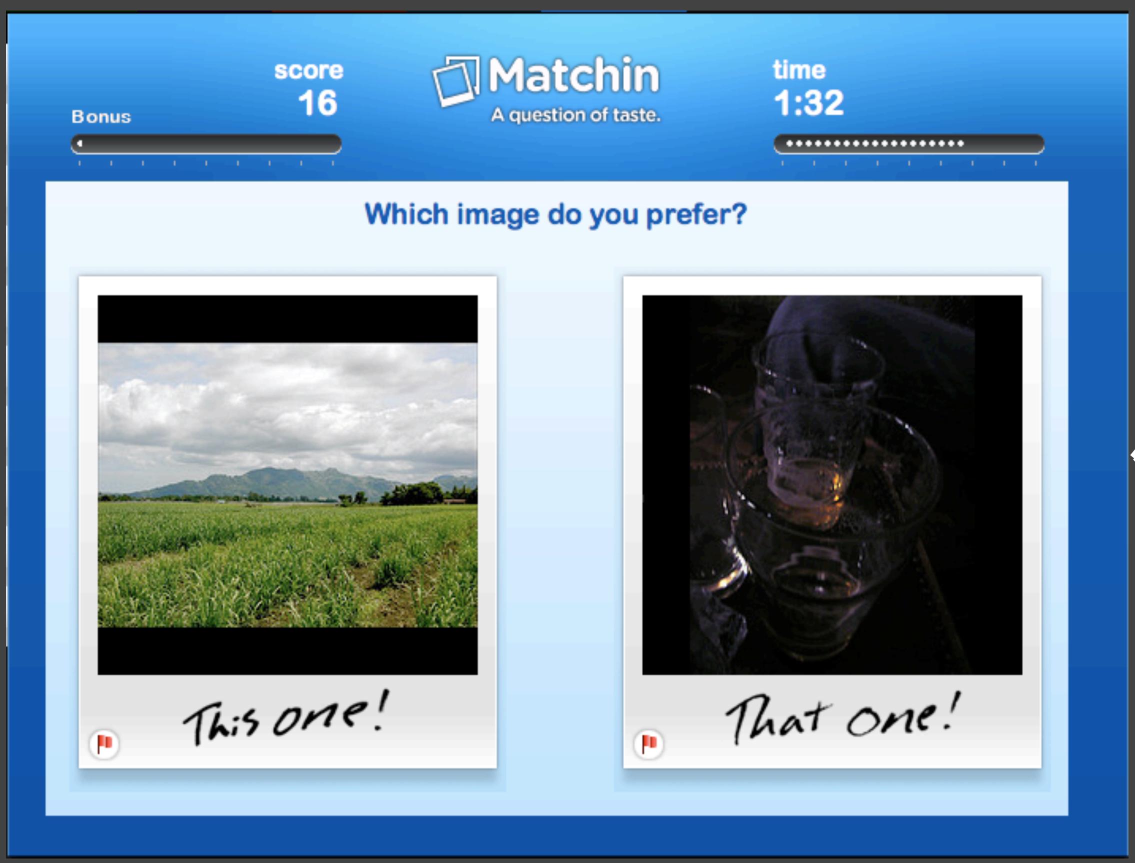
rating (scale 1-4)

ordering

(Cohen et al., 1999; Dwork et al., 2010; Ailon et al., 2005; Fagin et al., 2006)

Ranking Aggregation

an example



(Hacker et al., 2009)

Consensus Clustering

individual clusterings ➠ single clustering

(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)

Consensus Clustering

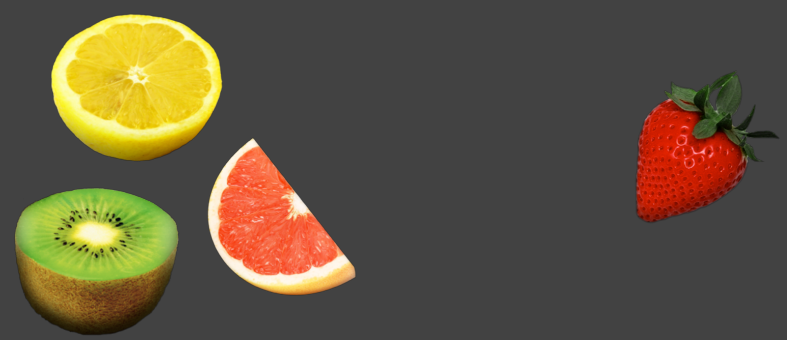
individual clusterings ➠ single clustering

complete clustering

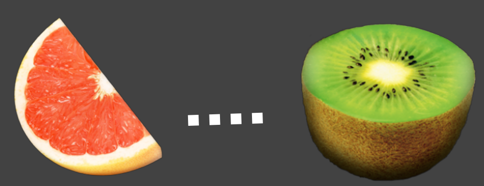
(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)

Consensus Clustering

individual clusterings ➠ single clustering



complete clustering

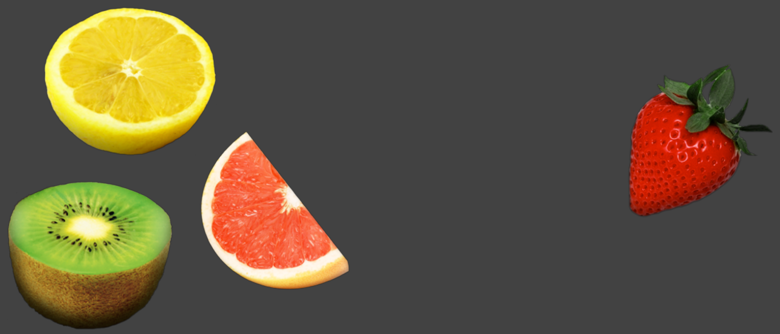


link / cannot-link constraints

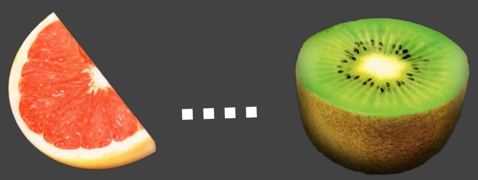
(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)

Consensus Clustering

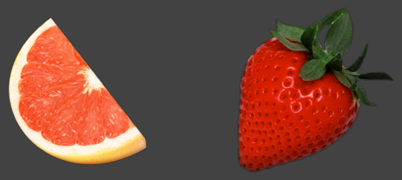
individual clusterings ➠ single clustering



complete clustering



link / cannot-link constraints



how similar are these? (scale 1-5)

(Topchy et al., 2005; Strehl and Ghosh 2003; Hu and Sung, 2006)

Consensus Clustering

an example

(Parent and Eskenazi, 2010)

Consensus Clustering

an example

Do the following definitions of the word aid have the same or different meaning?

• a piece of equipment that helps you to do something.

• something such as a machine or tool that helps someone do something.

LOCAL VIEW

(Parent and Eskenazi, 2010)

Consensus Clustering

an example

Do the following definitions of the word aid have the same or different meaning?

- a piece of equipment that helps you to do something.
- something such as a machine or tool that helps someone do something.

LOCAL VIEW

You have to group the definitions for the word 'code'. There are 2 general meanings.

- to mark a group of things with different colors so that you can tell the difference between them.
- to put a message in code so that it is secret.
- to put a set of numbers, letters, or signs on something to show that it is or give information about it,
- to represent a message in code so that it can only be understand by the person who is meant to receive it.

GLOBAL VIEW

(Parent and Eskenazi, 2010)

Challenge #2:

the correspondence problem

Structured Outputs

transcription, translation and description

least difficult

most difficult

Structured Outputs

transcription, translation and description

least difficult

↓

Transcription

- ROVER method (Fiscus, 1997)
- Longest Common Subsequences, Lattice (Evanini et al., 2010)

most difficult

Structured Outputs

transcription, translation and description

least difficult

↓

most difficult

Transcription

ROVER method (Fiscus, 1997)

Longest Common Subsequences, Lattice (Evanini et al., 2010)

Translation

BLEU (Pipineni et al., 2002);

Consensus Translation (Bangalore et al., 2001; Frederking and Nirenburg, 1994, Matusov et al., 2006, Rosti et al., 2007)

Structured Outputs

transcription, translation and description

least difficult

Transcription

ROVER method (Fiscus, 1997)

Longest Common Subsequences, Lattice (Evanini et al., 2010)

Translation

BLEU (Pipineni et al., 2002);

Consensus Translation (Bangalore et al., 2001; Frederking and Nirenburg, 1994, Matusov et al., 2006, Rosti et al., 2007)

Description

Information Fushion (Barzilay, 2003; Barzilay et al., 1999)

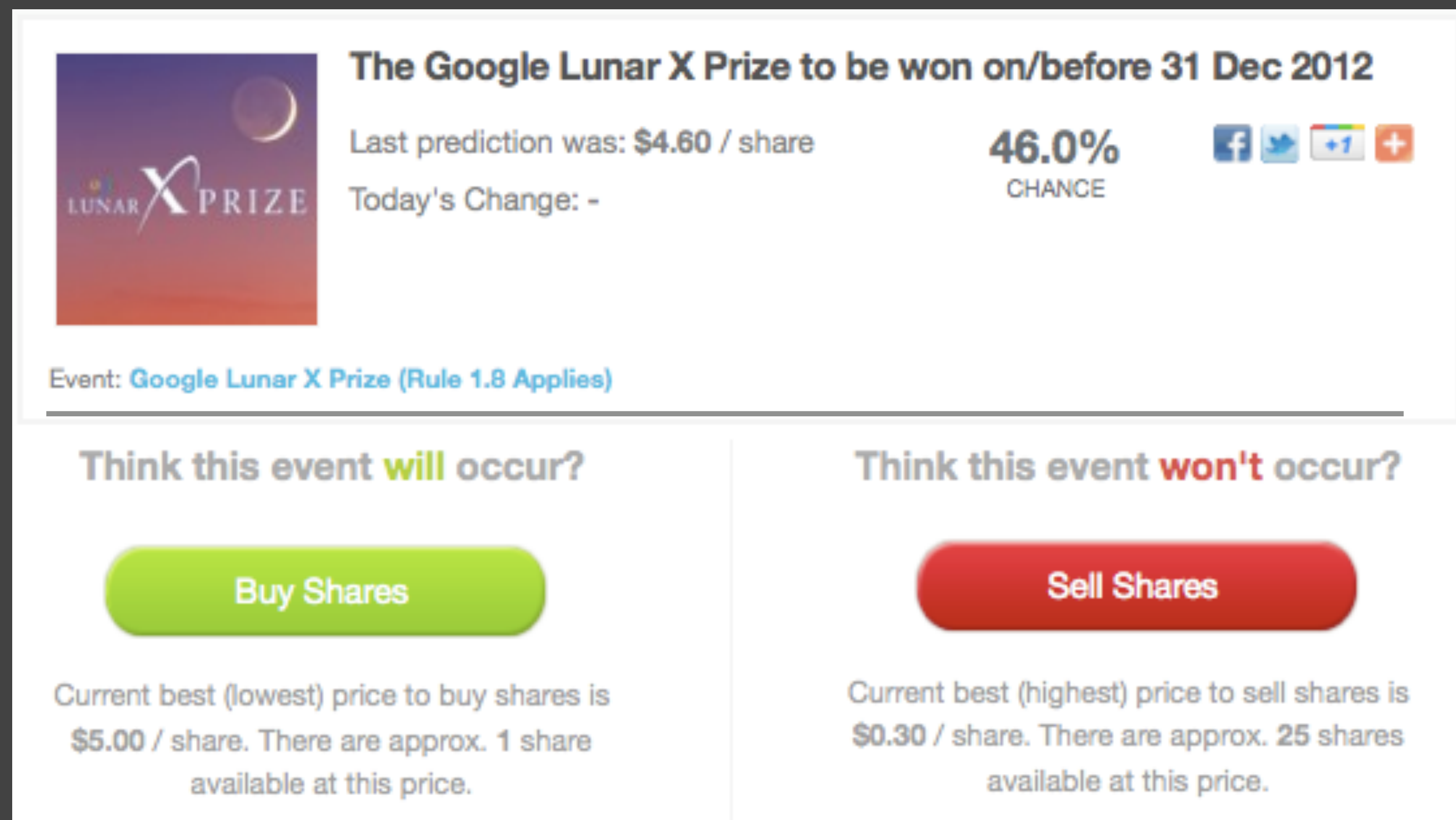
most difficult

Challenge #3:

aggregating difficult to articulate outputs

Belief Aggregation

with prediction markets



RECAP

TAKE-HOME

"classification and beyond"

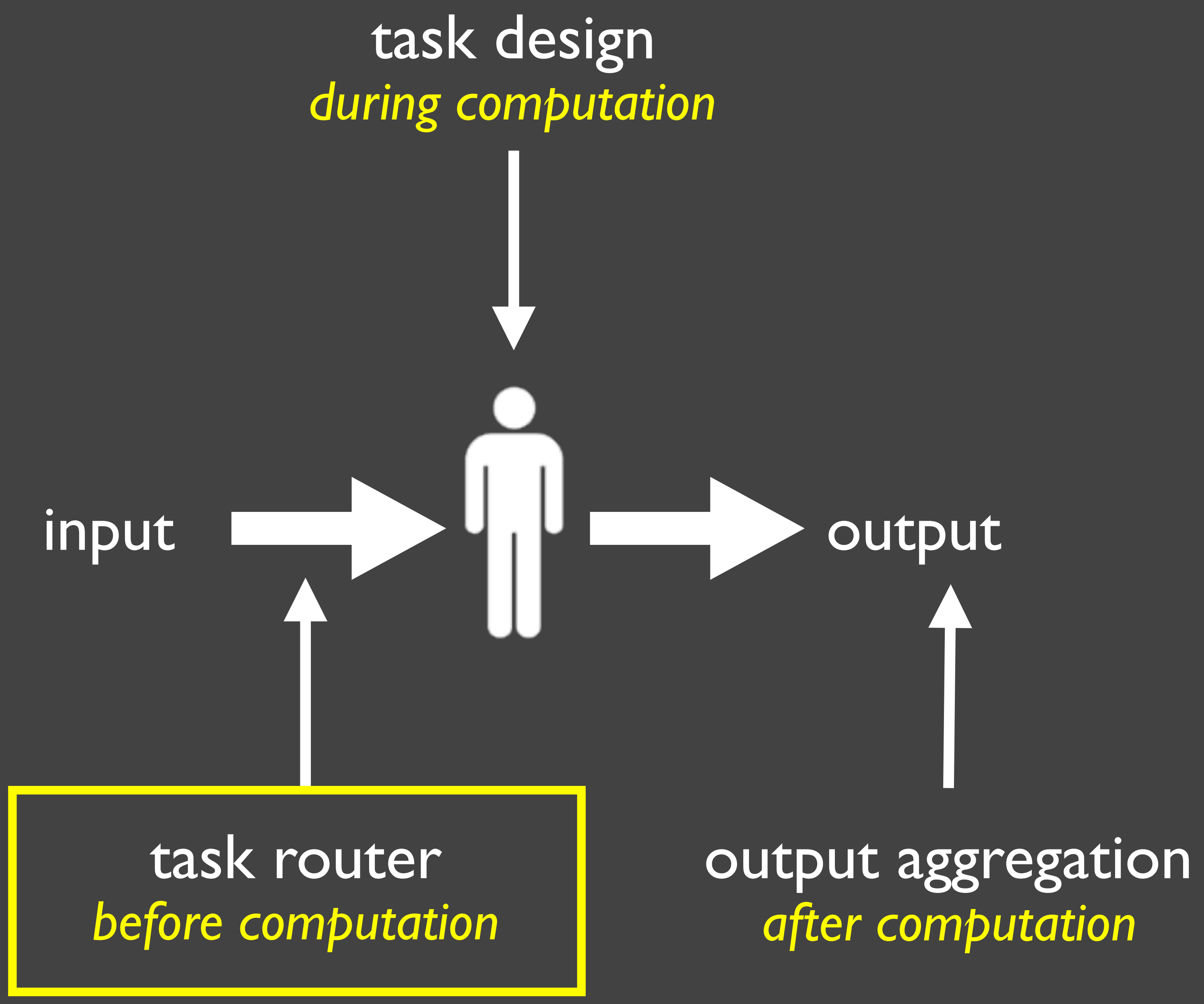
VI

TASK ROUTING

Motivation • Push Methods • Pull Methods

Correctness

Three Points of Intervention



MOTIVATION

The most popular task routing method is WHTBT.

The most popular task routing method is WHTBT.

(which stands for “Whoever Happens To Be There”).

All human computers are not created equal.

Push versus Pull

methods of task routing

Push versus Pull

methods of task routing

Push

Workers are passive receivers of tasks.

The system takes complete control over who is assigned which task.

Push versus Pull
methods of task routing

Push

Workers are passive receivers of tasks.

The system takes complete control over who is assigned which task.

Pull

Workers are active seekers of tasks.

The system supports a set of interfaces that enable workers to look for tasks to assign themselves.

PUSH METHODS

Push Methods
system ➠ workers

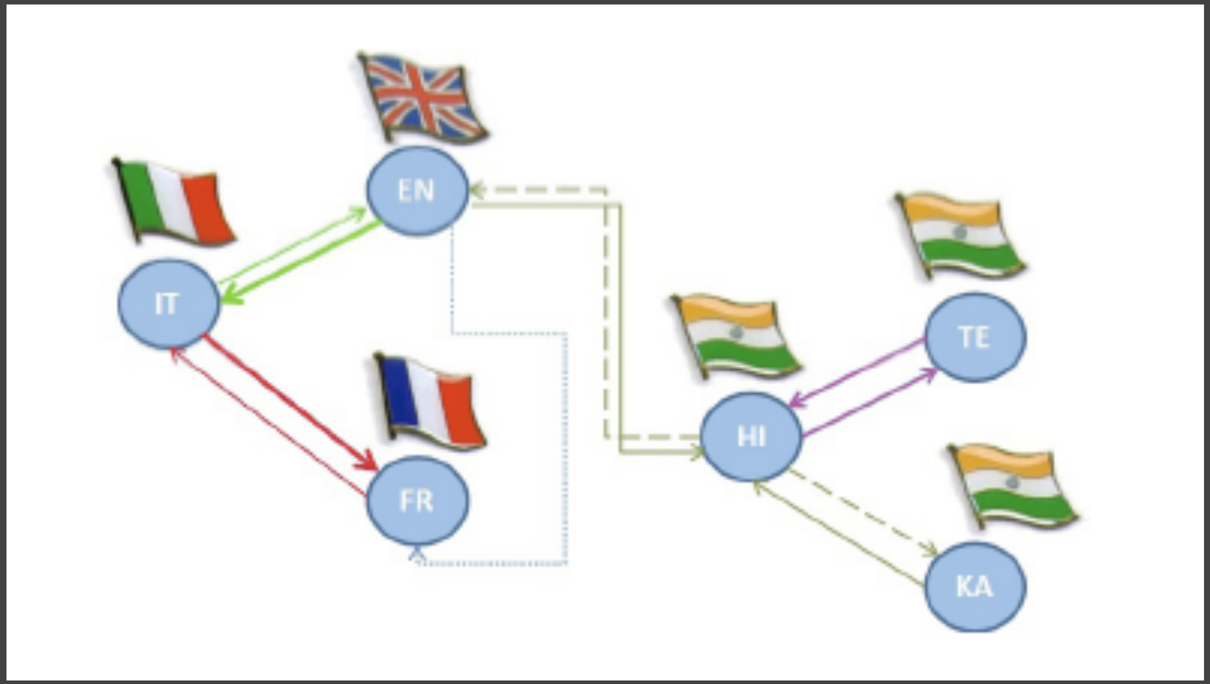
The system takes complete control over who is assigned which task.

Allocation

complete knowledge of utility and cost

Allocation

complete knowledge of utility and cost

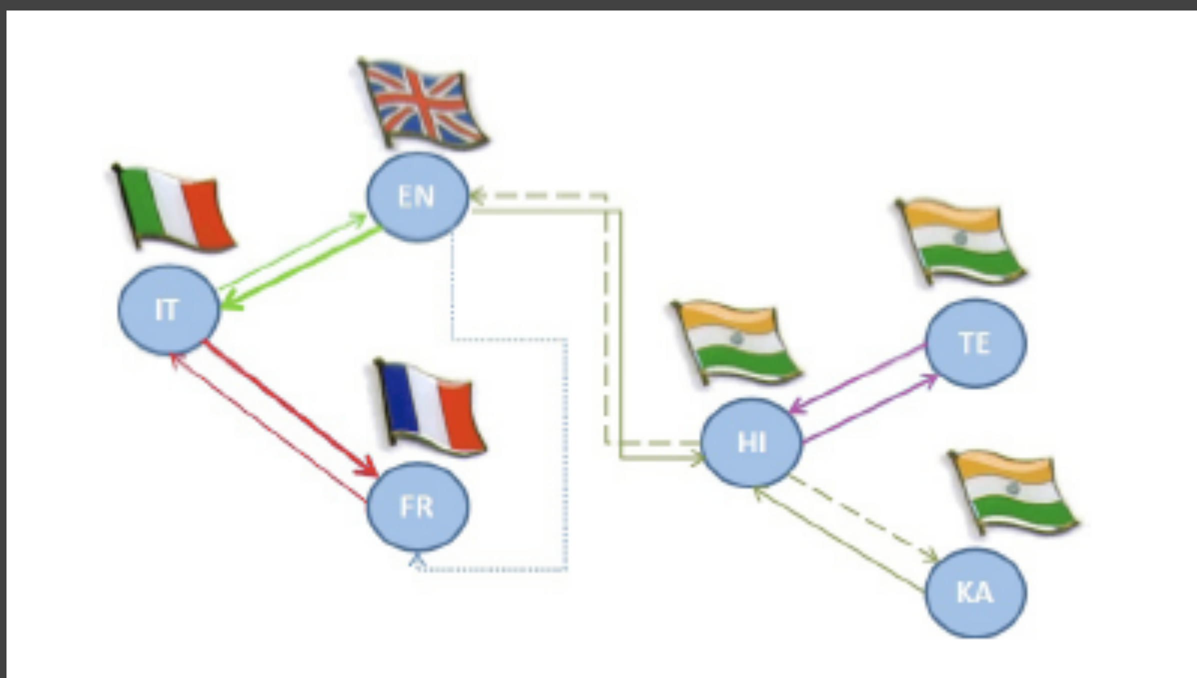


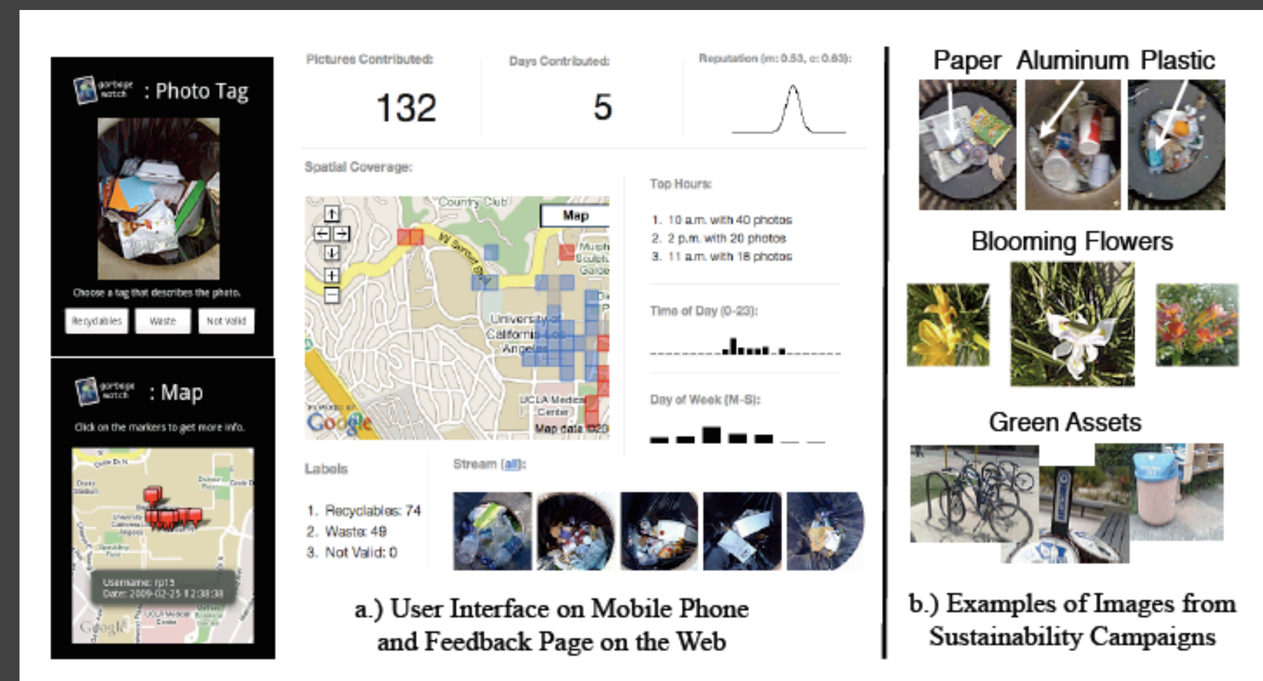
(Shahaf and Horvitz, 2010)
Workers have known competencies.
Tasks have known demands.

weighted exact set-cover problem

Allocation

complete knowledge of utility and cost





a.) User Interface on Mobile Phone and Feedback Page on the Web

b.) Examples of Images from Sustainability Campaigns

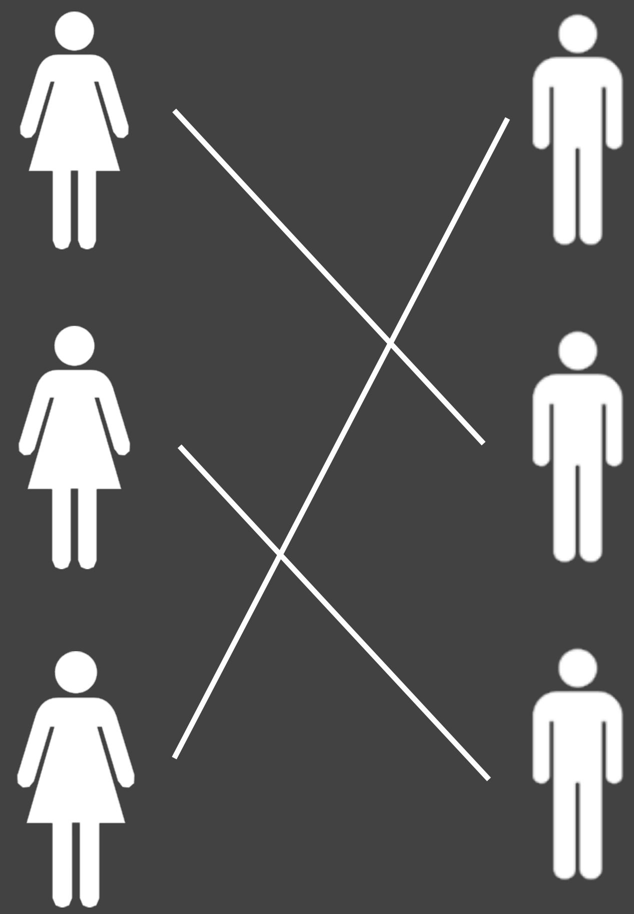
(Shahaf and Horvitz, 2010)
Workers have known competencies.
Tasks have known demands.

weighted exact set-cover problem

(Reddy et al., 2010)
Participants have known cost and utility (based on what they can cover).

budgeted maximum coverage problem

Matching
complete or partial preferences



man to woman (Gale and Shapley, 1962)
medical residents to hospitals (Roth, 1984)
students to schools (Teo, 2001)
sailors to ships (Liebowitz, 2000)

incomplete information (Gusfield and Irving, 1989; Liebowitz, 2000)

Inference

incomplete knowledge of utility and cost

Inference
incomplete knowledge of utility and cost

Decision-Theoretic Model

e.g., Donmez et al., 2008

Discovery and Assignment Phase

Inference

incomplete knowledge of utility and cost

Decision-Theoretic Model

e.g., Donmez et al., 2008

Discovery and Assignment Phase

Online Learning

e.g., Donmez et al., 2009

Exploration-Exploitation Tradeoff

PULL METHODS

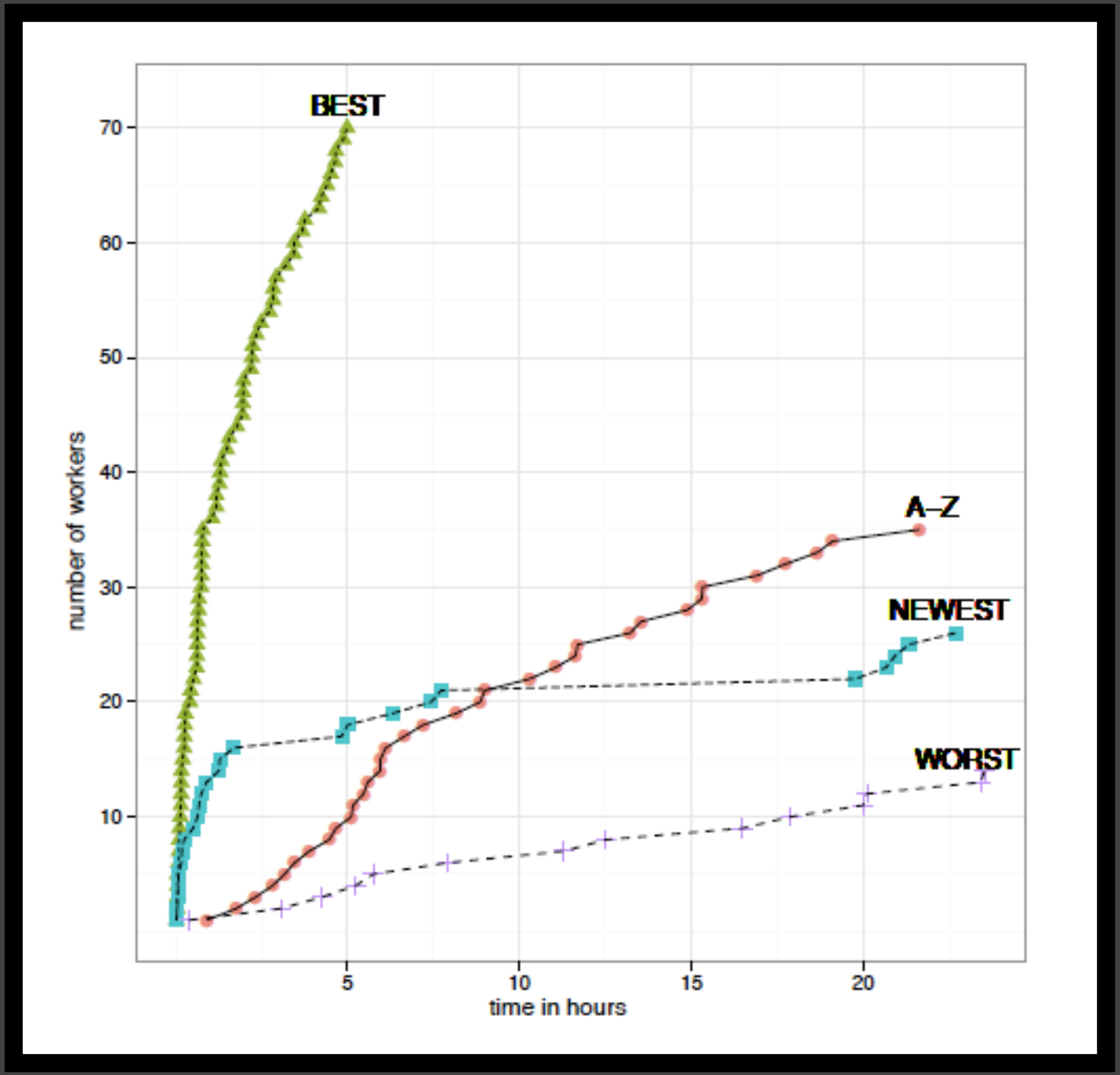
Pull Methods

workers ➠ system

The system merely sets up the environment to allow workers to assign themselves (or each other) tasks.

Search

locating tasks



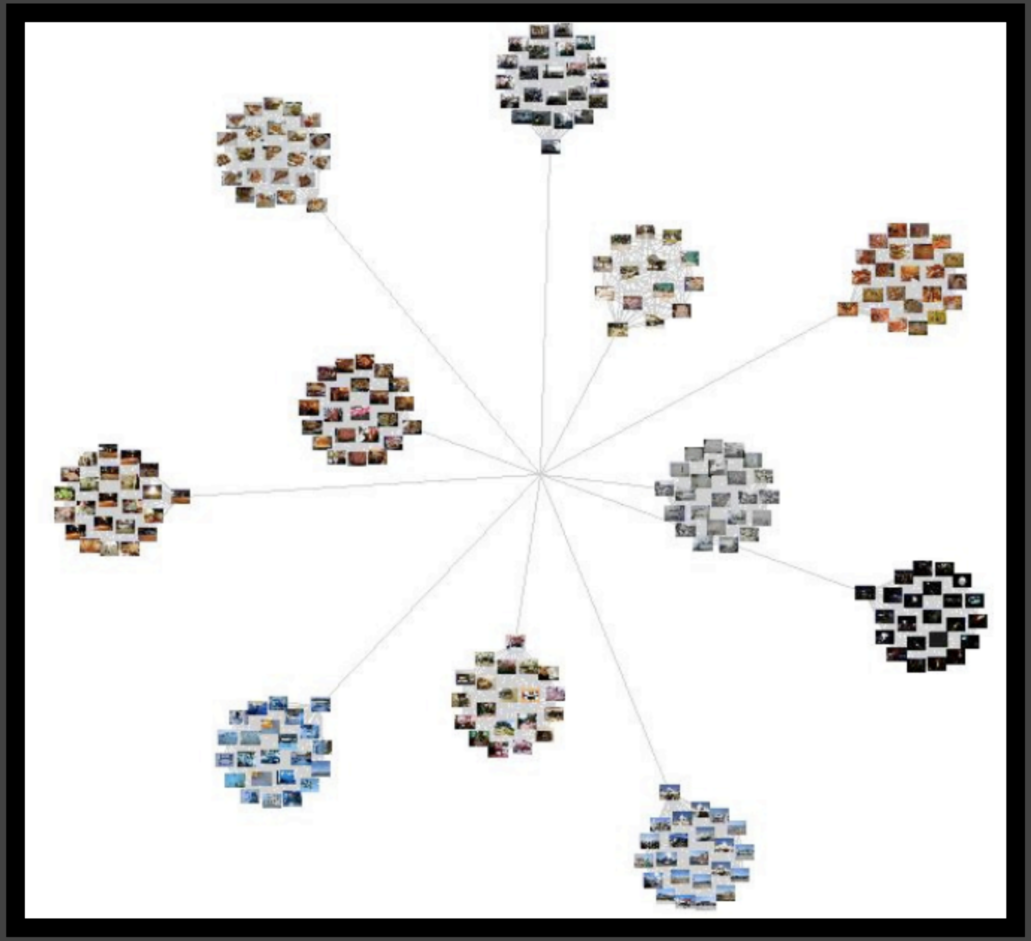
(Chilton et al., 2010)

Visualization

locating particular input objects

Visualization

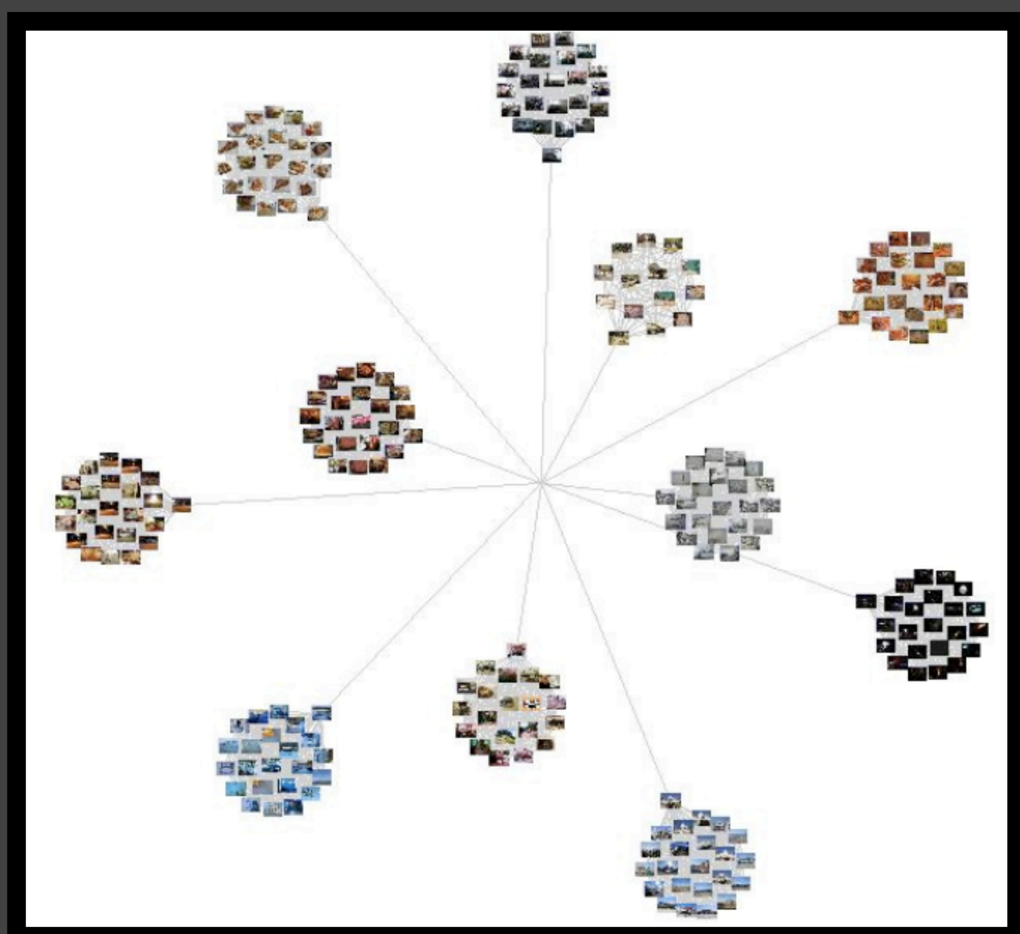
locating particular input objects



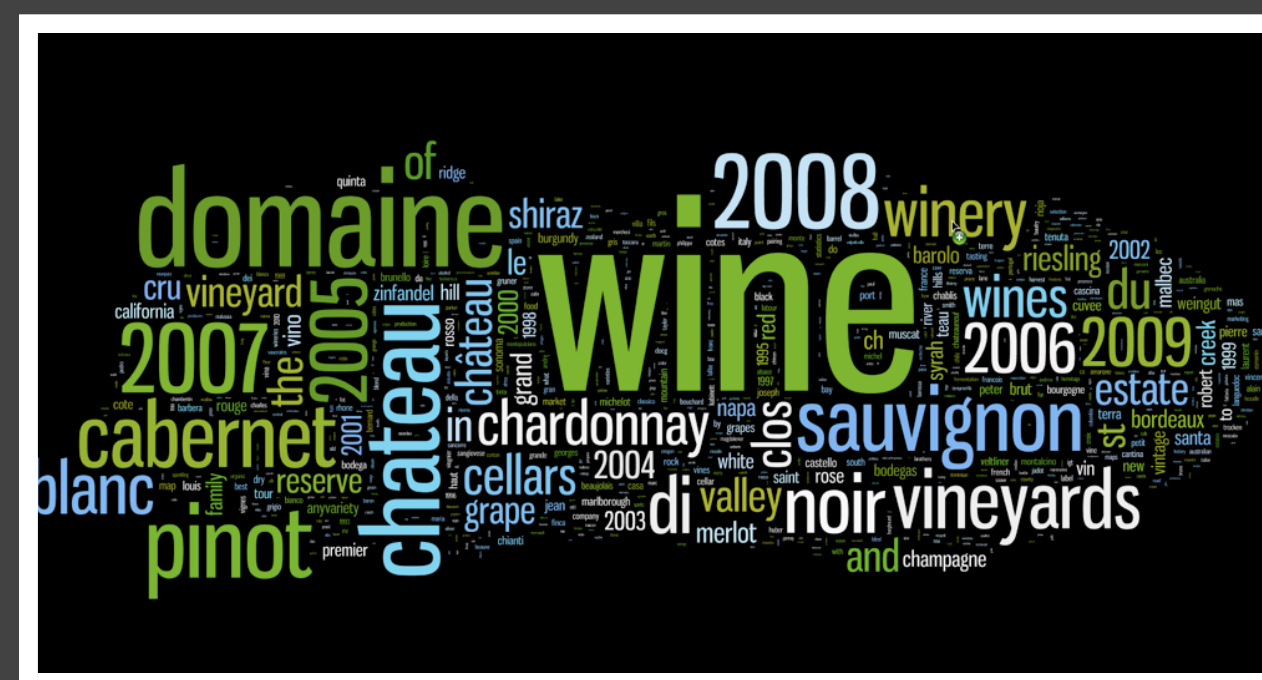
(Borden, 2006)

Visualization

locating particular input objects



(Borden, 2006)



source: ablegrape.wordpress.com

Task Recommendation

personalization

Task Recommendation
personalization

Content-Based
find similarities between worker profile and task characteristics.

Collaborative Filtering
make use of preference information about tasks (e.g., ratings) to infer similarities between workers.

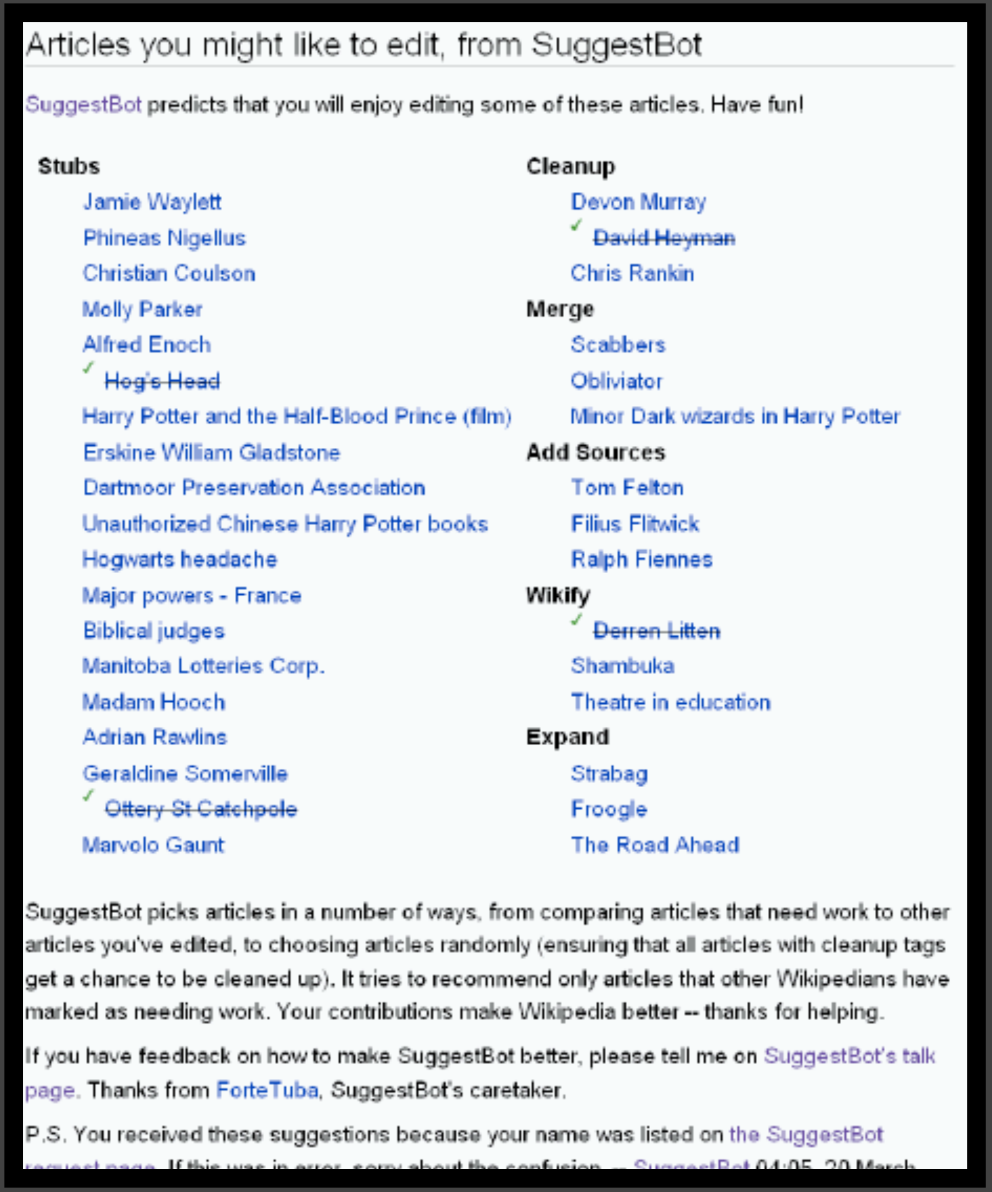
Hybrid
a mix of content-based and collaborative filtering methods.

Task Recommendation
personalization

Content-Based
find similarities between worker profile and task characteristics.

Collaborative Filtering
make use of preference information about tasks (e.g., ratings) to infer similarities between workers.

Hybrid
a mix of content-based and collaborative filtering methods.



SuggestBot (Cosley et al., 2006)

Peer Routing

people's knowledge of each other

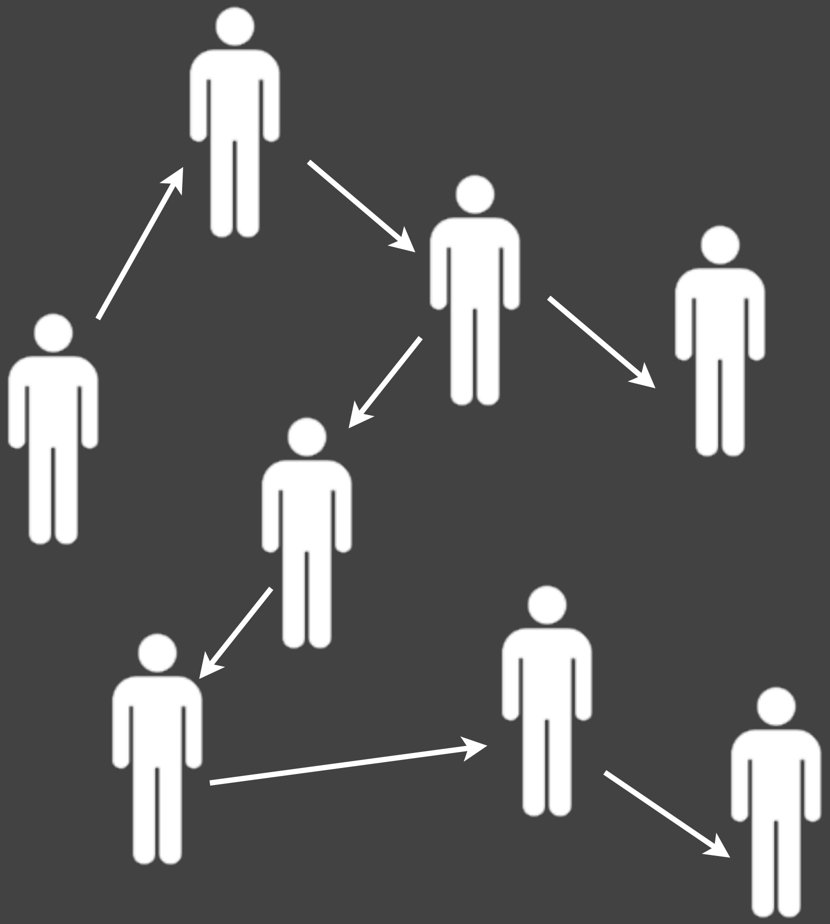
Peer Routing

people's knowledge of each other

DARPA Red Balloon Challenge

Peer Routing
people's knowledge of each other

DARPA Red Balloon Challenge



(Zhang et al., 2011)

RECAP

TAKE-HOME

"Wisdom of the individuals in the crowd"

VII

CONCLUSION

Conclusion

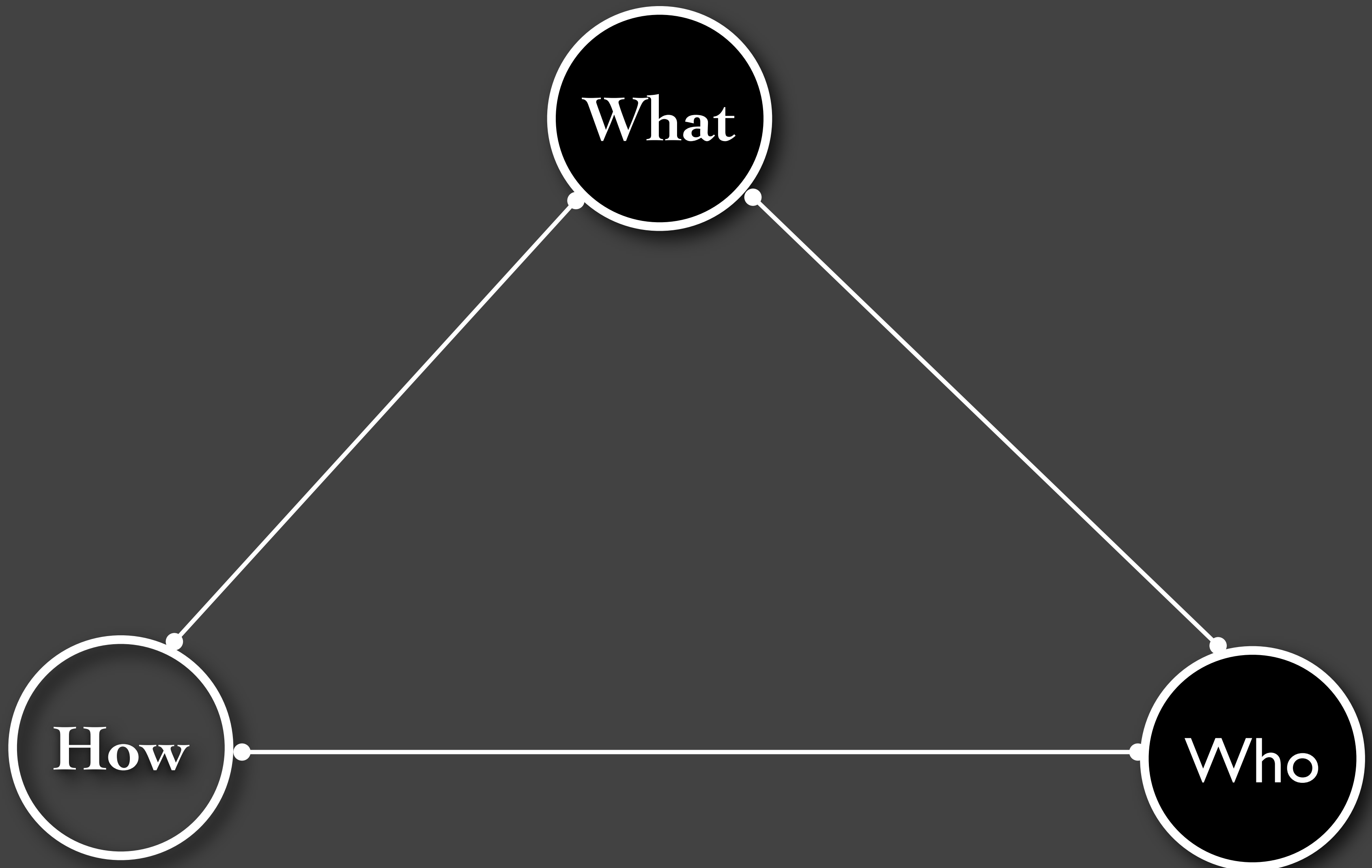
Summary
What have we learned?

Closing
What are some opportunities for AI research?

SUMMARY

Bird's Eye View

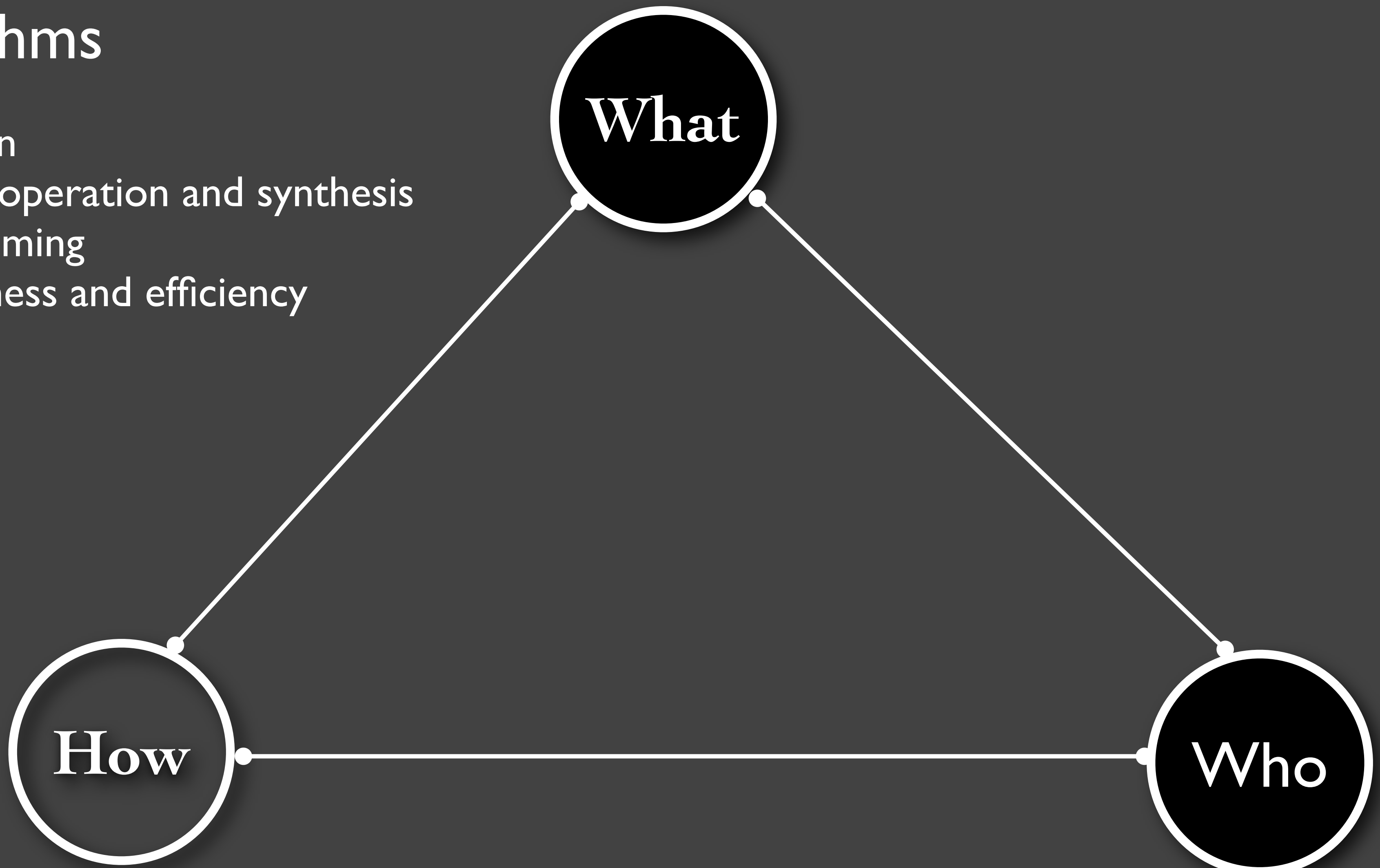
of this tutorial



Bird's Eye View
of this tutorial

Human Computation Algorithms

- definition
- control, operation and synthesis
- programming
- correctness and efficiency



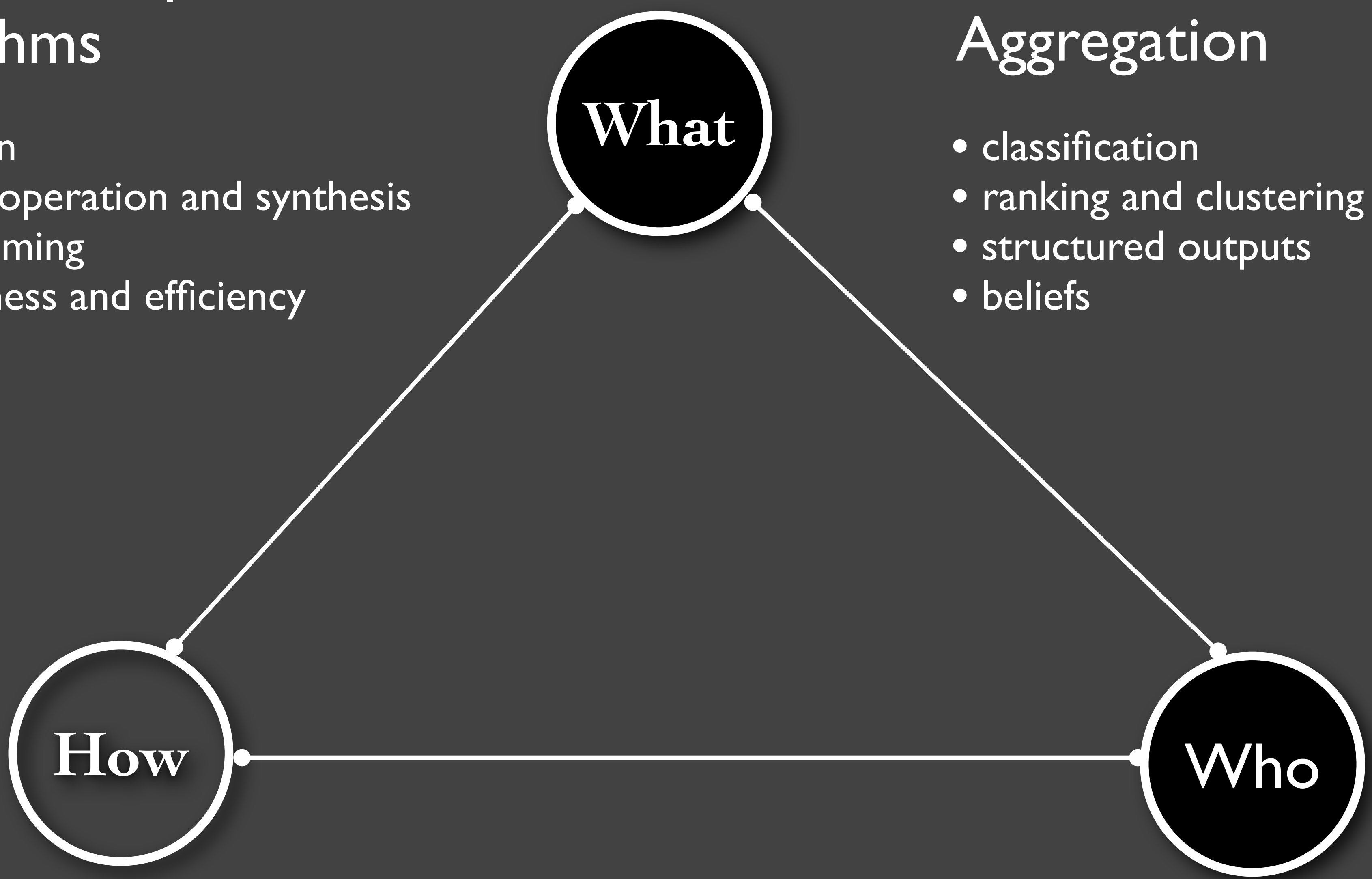
Bird’s Eye View
of this tutorial

Human Computation Algorithms

- definition
- control, operation and synthesis
- programming
- correctness and efficiency

Output Aggregation

- classification
- ranking and clustering
- structured outputs
- beliefs



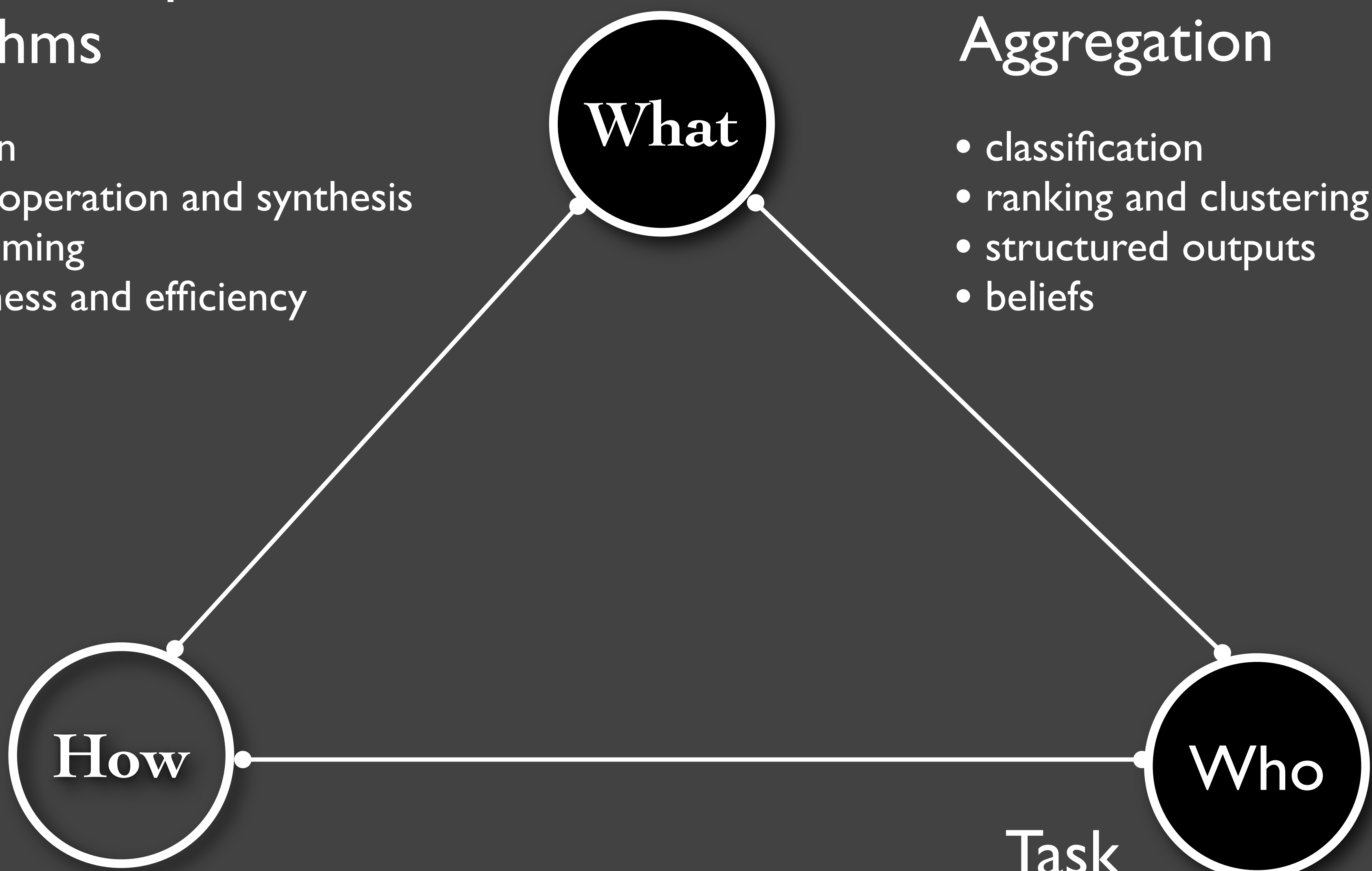
Bird's Eye View
of this tutorial

Human Computation Algorithms

- definition
- control, operation and synthesis
- programming
- correctness and efficiency

Output Aggregation

- classification
- ranking and clustering
- structured outputs
- beliefs

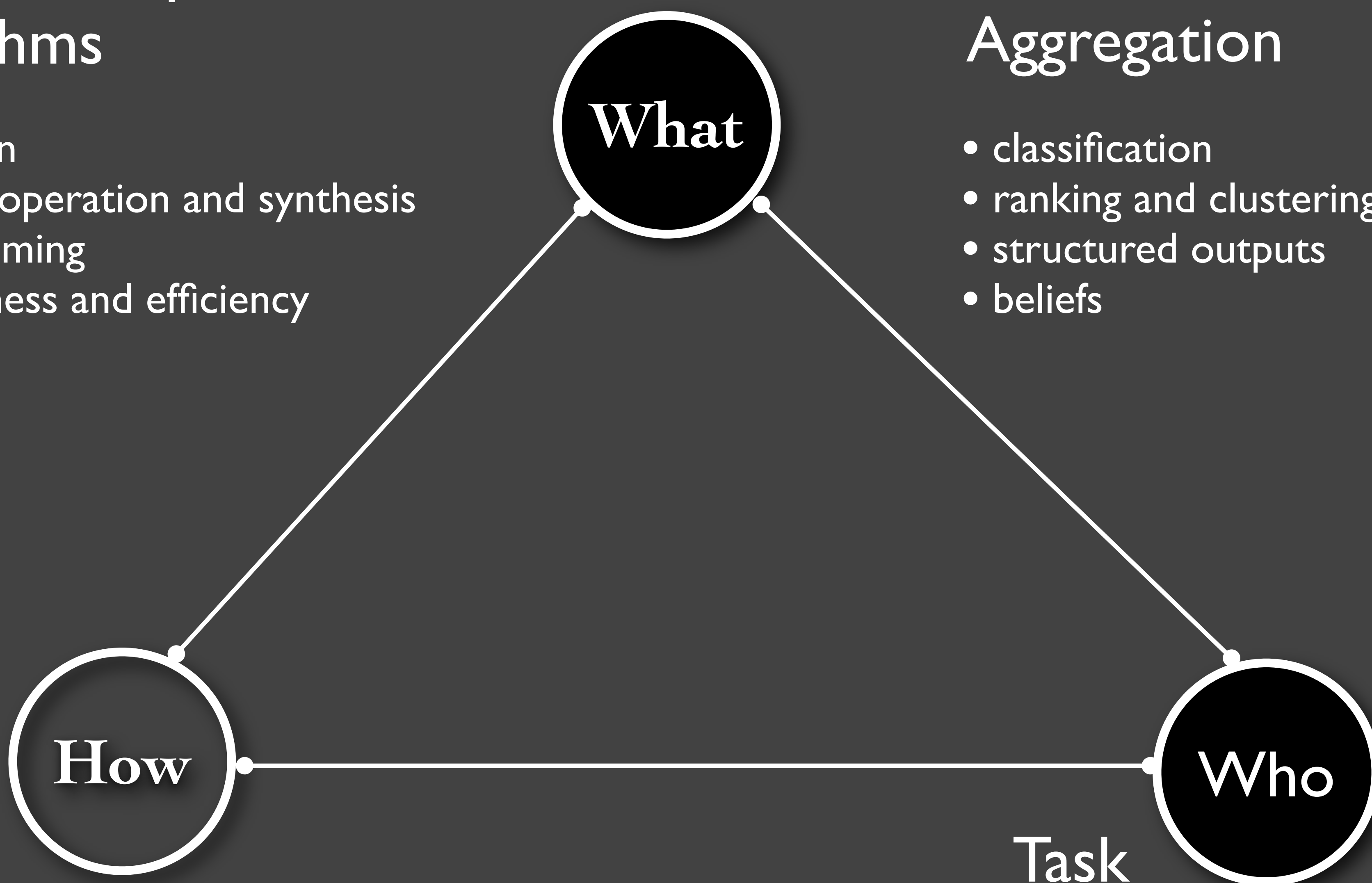


Task Routing

- push versus pull
- allocation / matching
- inference / online learning

Bird's Eye View

of this tutorial



Human Computation Algorithms

- definition
- control, operation and synthesis
- programming
- correctness and efficiency

Output Aggregation

- classification
- ranking and clustering
- structured outputs
- beliefs

Designing for Human Computers

- who they are
- what are their wants and needs

The Art of Asking Questions

- task design
- game design

Task Routing

- push versus pull
- allocation / matching
- inference / online learning

CLOSING

The Role of AI

in human computation

The Role of AI
in human computation

AI as requesters

learning to recognize objects, translate sentences, classifying music by querying human teachers.

The Role of AI
in human computation

AI as requesters
learning to recognize objects, translate sentences, classifying music by querying human teachers.

AI as optimizers
improve the accuracy and efficiency of human computation algorithms.

The Role of AI
in human computation

AI as requesters
learning to recognize objects, translate sentences, classifying music by querying human teachers.

AI as optimizers
improve the accuracy and efficiency of human computation algorithms.

AI as enablers
make human computers better, e.g., by organizing and displaying information to workers.

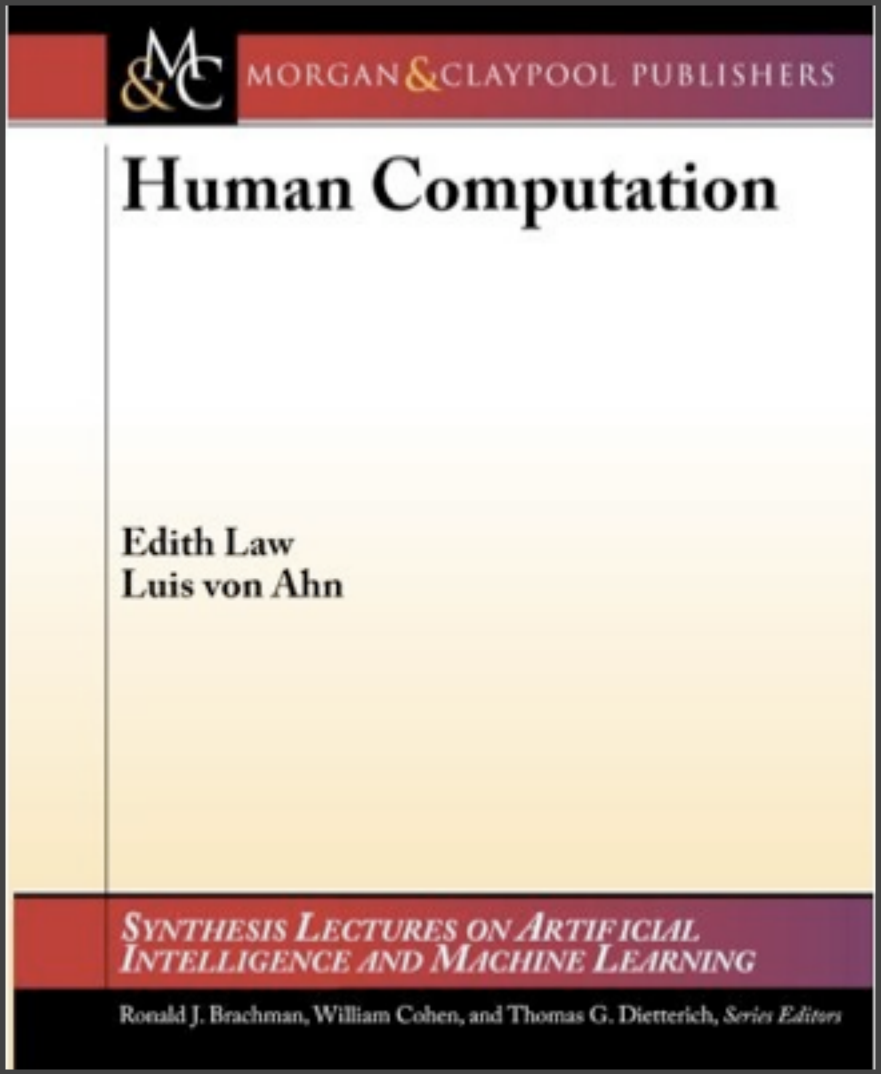
The Role of AI
in human computation

AI as requesters
learning to recognize objects, translate sentences, classifying music by querying human teachers.

AI as optimizers
improve the accuracy and efficiency of human computation algorithms.

AI as enablers
make human computers better, e.g., by organizing and displaying information to workers.

AI as workers
perform tasks that they are better at than humans.



a conceptual framework

an annotated bibliography

a place to get ideas for research

a work in progress

free for you! Come pick one up during the break.

Other resources:
http://humancomputation.com/book

THANK YOU &
CATCH YOU @ COFFEE!