Semantic Technology in Watson

Chris Welty
IBM Research
ibmwatson.com

What is Watson?

- Open Domain Question-Answering Machine
- Given
  - Rich **Natural Language Questions**
  - Over a **Broad Domain of Knowledge**
- Delivers
  - **Precise Answers:** Determine what is being asked & give precise response
  - **Accurate Confidences:** Determine likelihood answer is correct
  - **Consumable Justifications:** Explain why the answer is right
  - **Fast Response Time:** Precision & Confidence in <3 seconds
  - At the level of human experts
- Proved its mettle in a televised match
  - Won a 2-game Jeopardy match against the all-time winners
  - viewed by over 50,000,000

2

The Jeopardy! Challenge

Hard for humans, hard for machines

IBM WATSON

Broad/Open Domain

Complex Language

High Precision

Accurate Confidence

High Speed

$200
If you are looking at the wainscoting, you are looking in this direction.
What is down?

$1000
The first person mentioned by name in 'The Man in the Iron Mask' is this hero of a
Who is D'Artagnan?

But hard for different reasons.

For people, the challenge is *knowing the answer*

For machines, the challenge is *understanding the question*

$600
In cell division, mitosis splits the nucleus &
What is cytoplasm?

$800
The conspirators against this man were wounded by
Who is Julius Caesar?

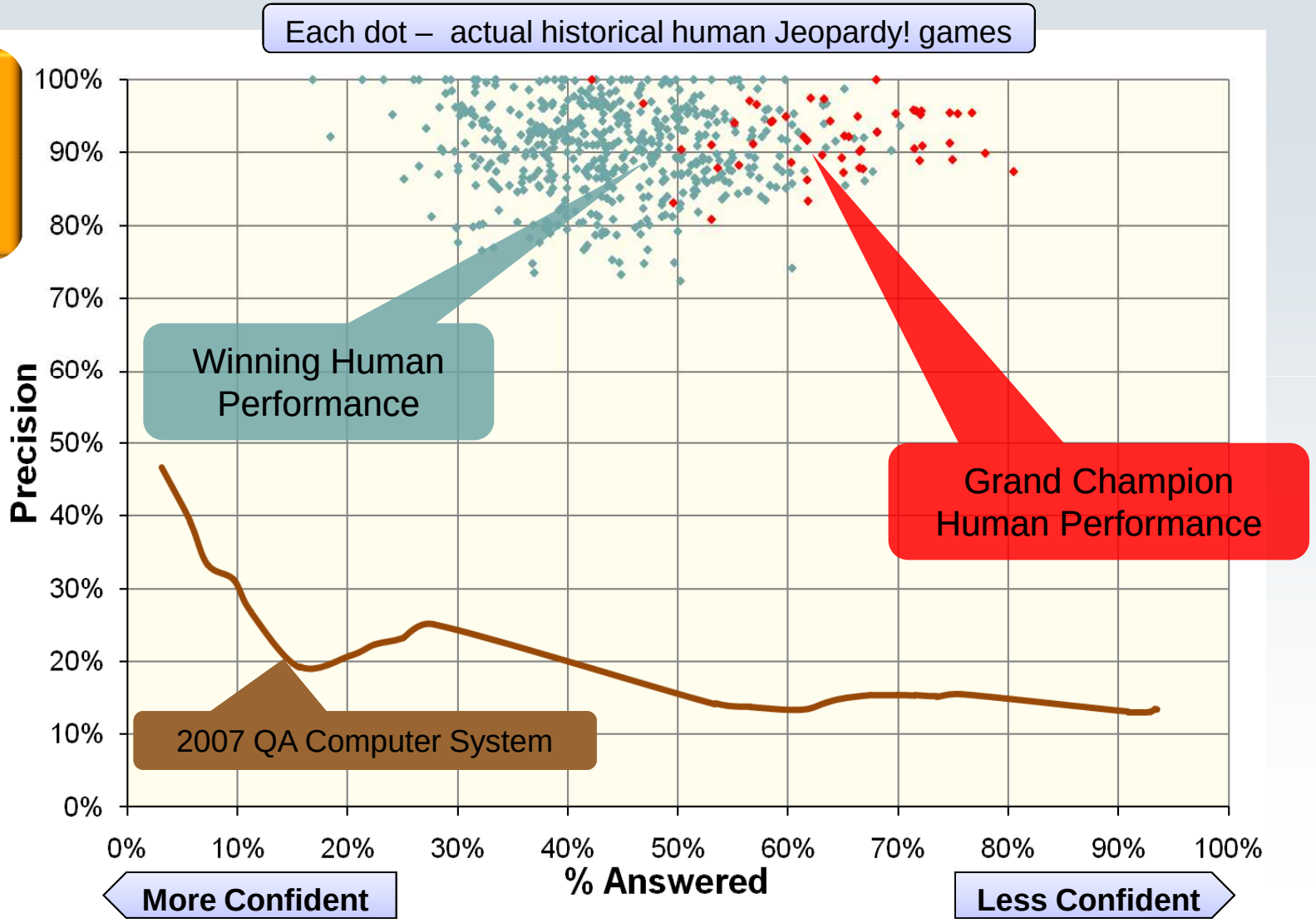
The Watson Enablers

- **J!-Archive**: the complete 30-year history of the Jeopardy! show, every question, every answer (200,000 questions)
- ✓ **Wikipedia**: 92% of Jeopardy questions have answers in Wikipedia. 81% of the answers are Wikipedia titles
- ✓ **Metric**: after a 4-month study of the problem we devised a satisfactory metric that we strongly believed was indicative of our chances of winning a game
- **Machine Learning**, ensemble methods: The emergence, availability, and stability of software for ML
- ✓ **Semantic Web**, linked data
- …and now we realize
  - *Jeopardy! questions have only one correct answer*
  - *Almost every answer is expressed in one sentence*

*The **Winner's Cloud***

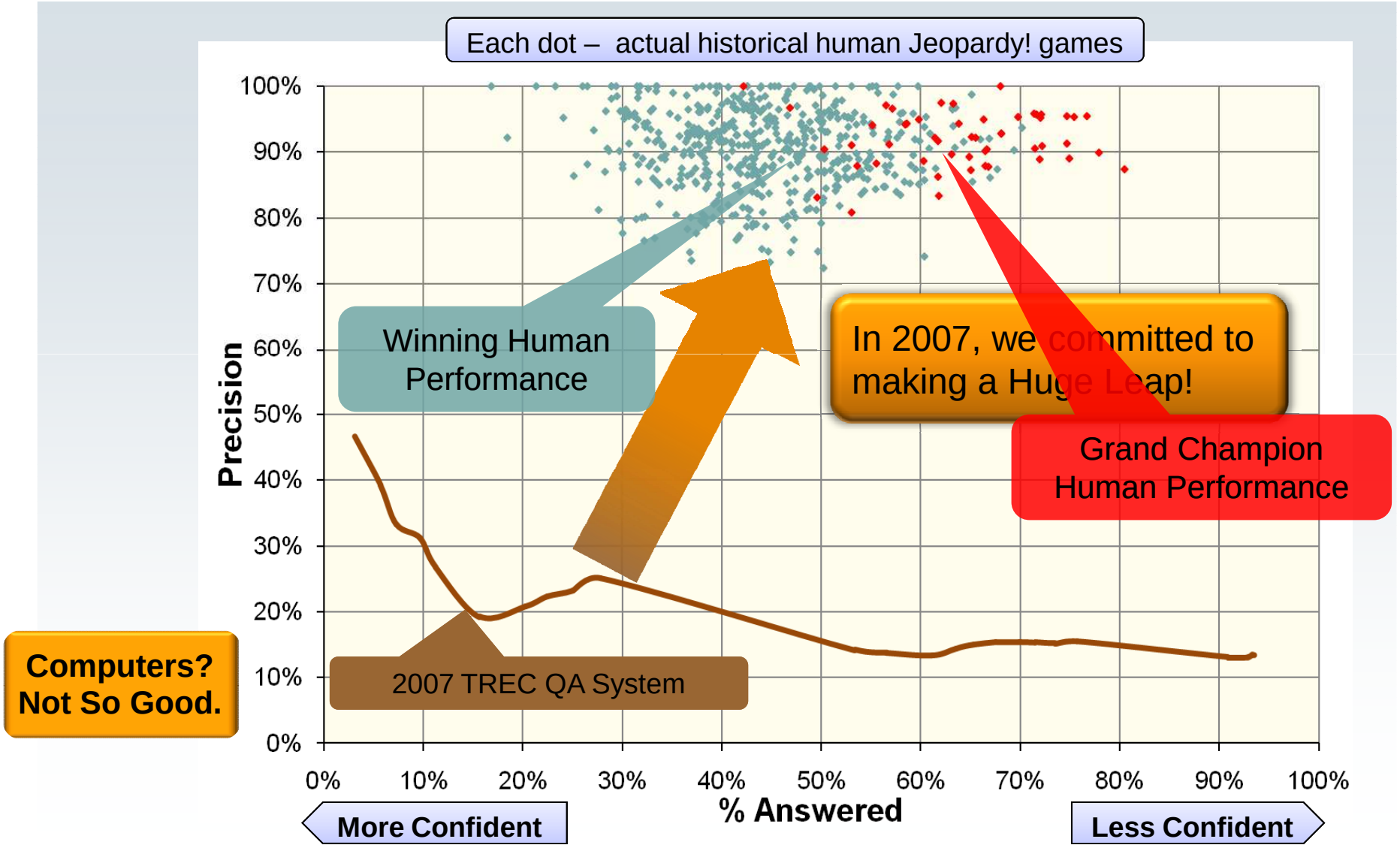
What It Takes to compete against Top Human Jeopardy! Players

Top human players are *remarkably* good.

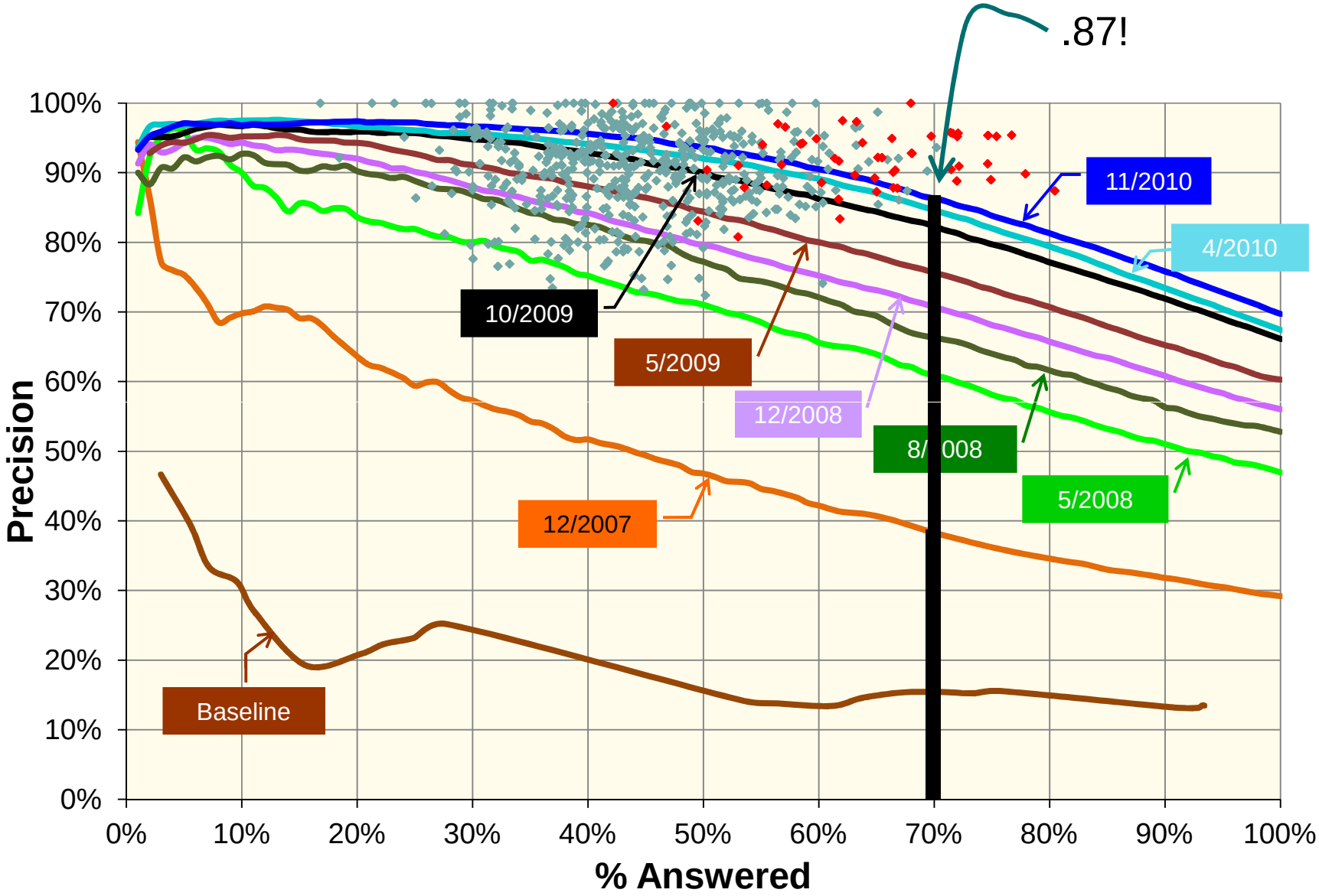


*The **Winner's Cloud***

What It Takes to compete against Top Human Jeopardy! Players



Steady Progress



What we do & don't

In May 1898 Portugal celebrated the 400th anniversary of this explorer's arrival in India.

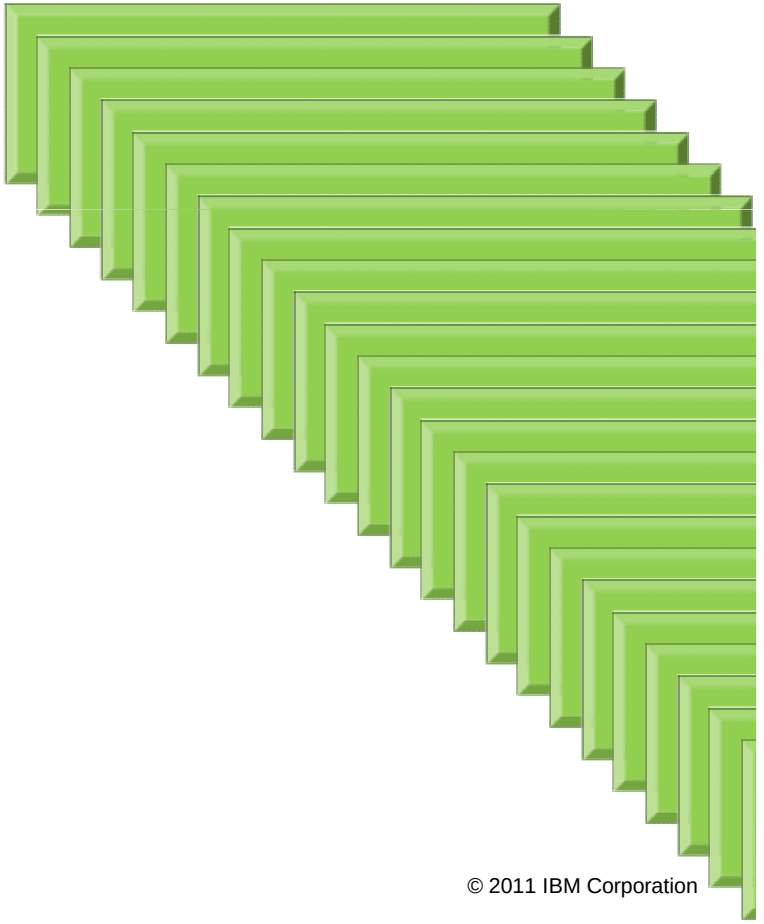
On the 27th of May 1498, Vasco da Gama landed in Kappad Beach

In May, Gary arrived in India after he celebrated his anniversary in Portugal.

DO: Find relevant passages that may contain an answer

DO: Match or align the passage with the question using diverse evidence

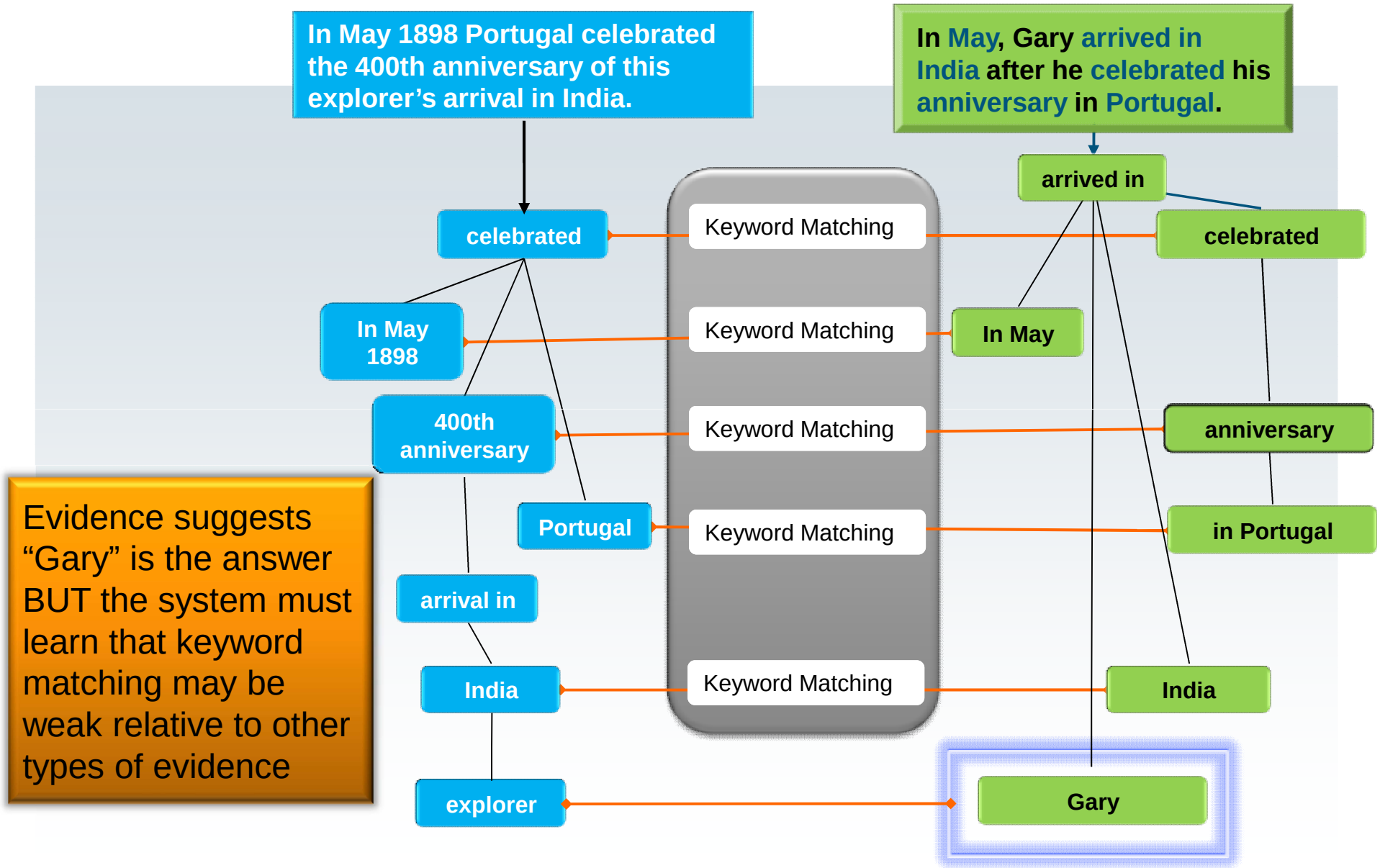
DON'T: Translate the question into a formal query and look up the answer



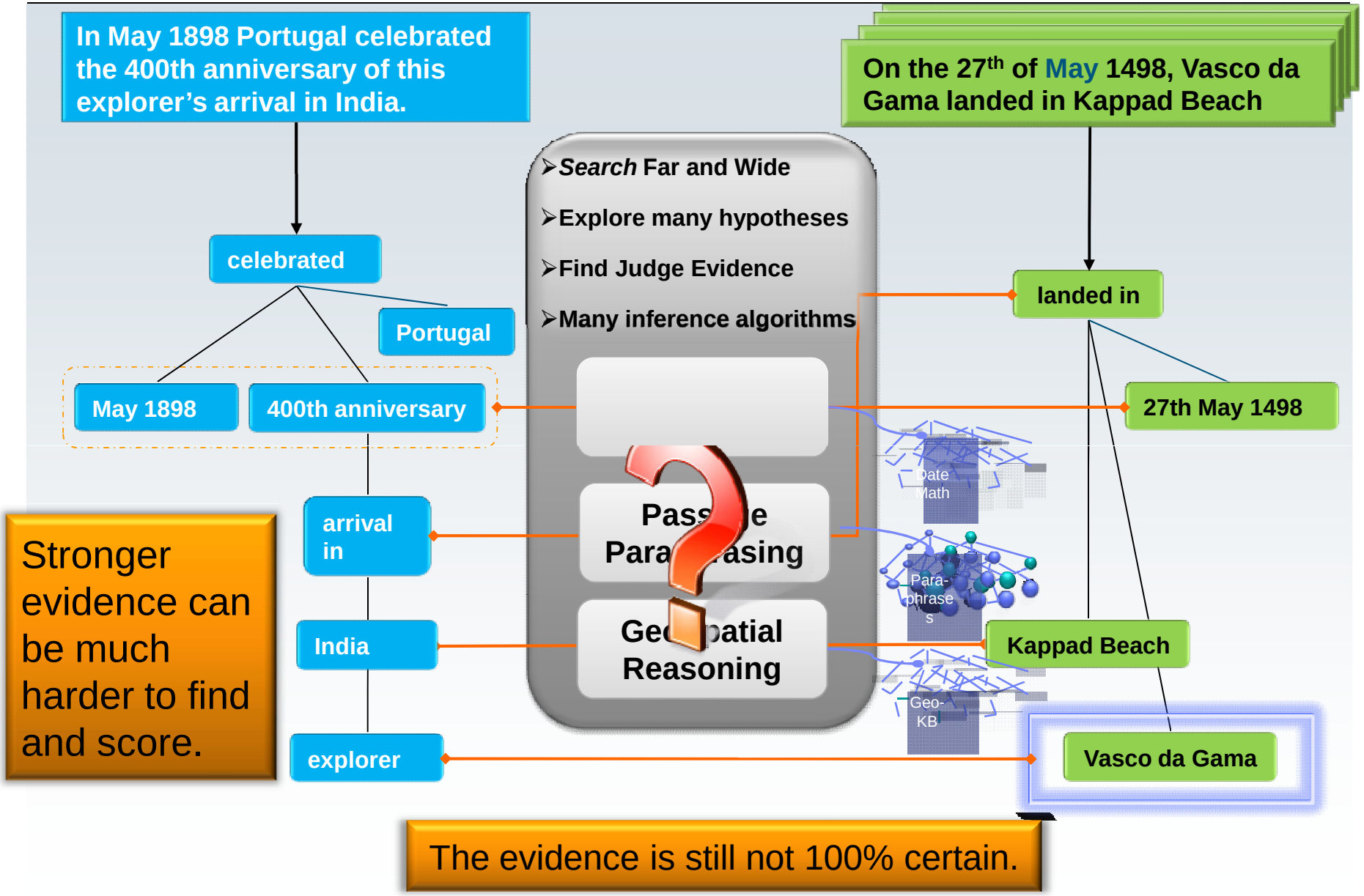
Celebration(e), date(e,1898), celebrationOf(e,e1), location(e, Portugal), date(e1, dateOf(e) – 400)), arrival(e1), location(e1,India), particpantOf(e1,?x).

Location(e2, Kappad Beach), Date(e2, 1498), landing(e2), particpantOf(e1,Vasco).

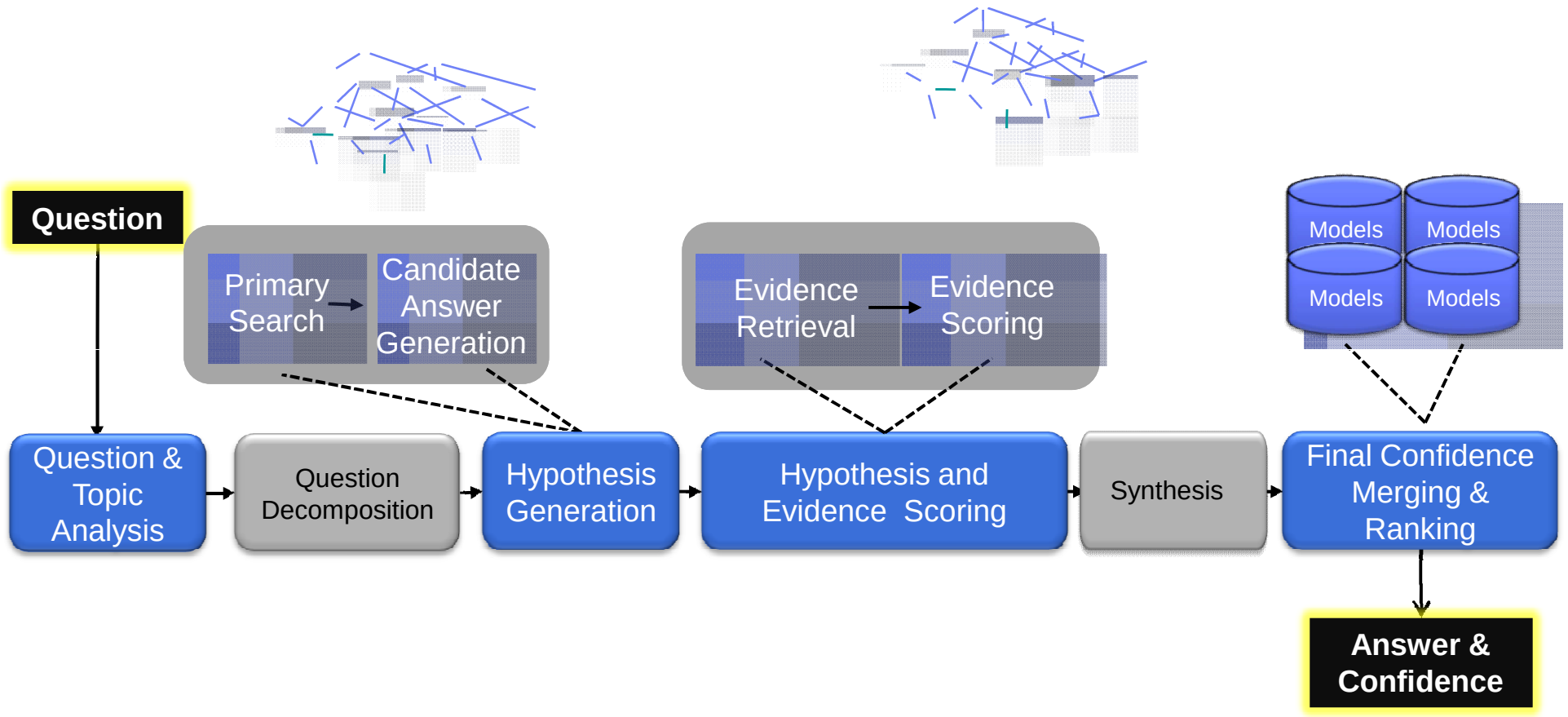
Matching Keyword Evidence



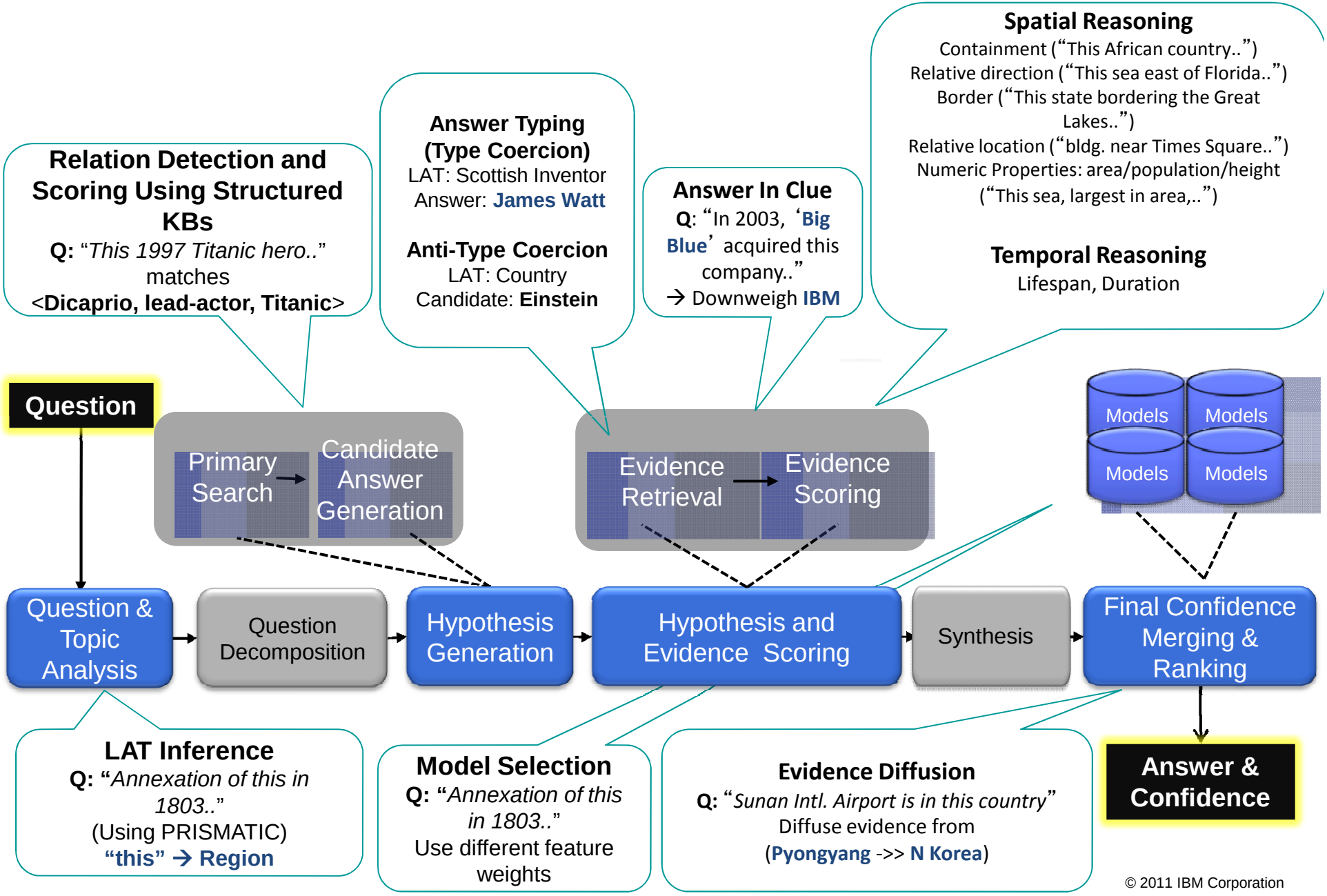
Matching Deeper Evidence



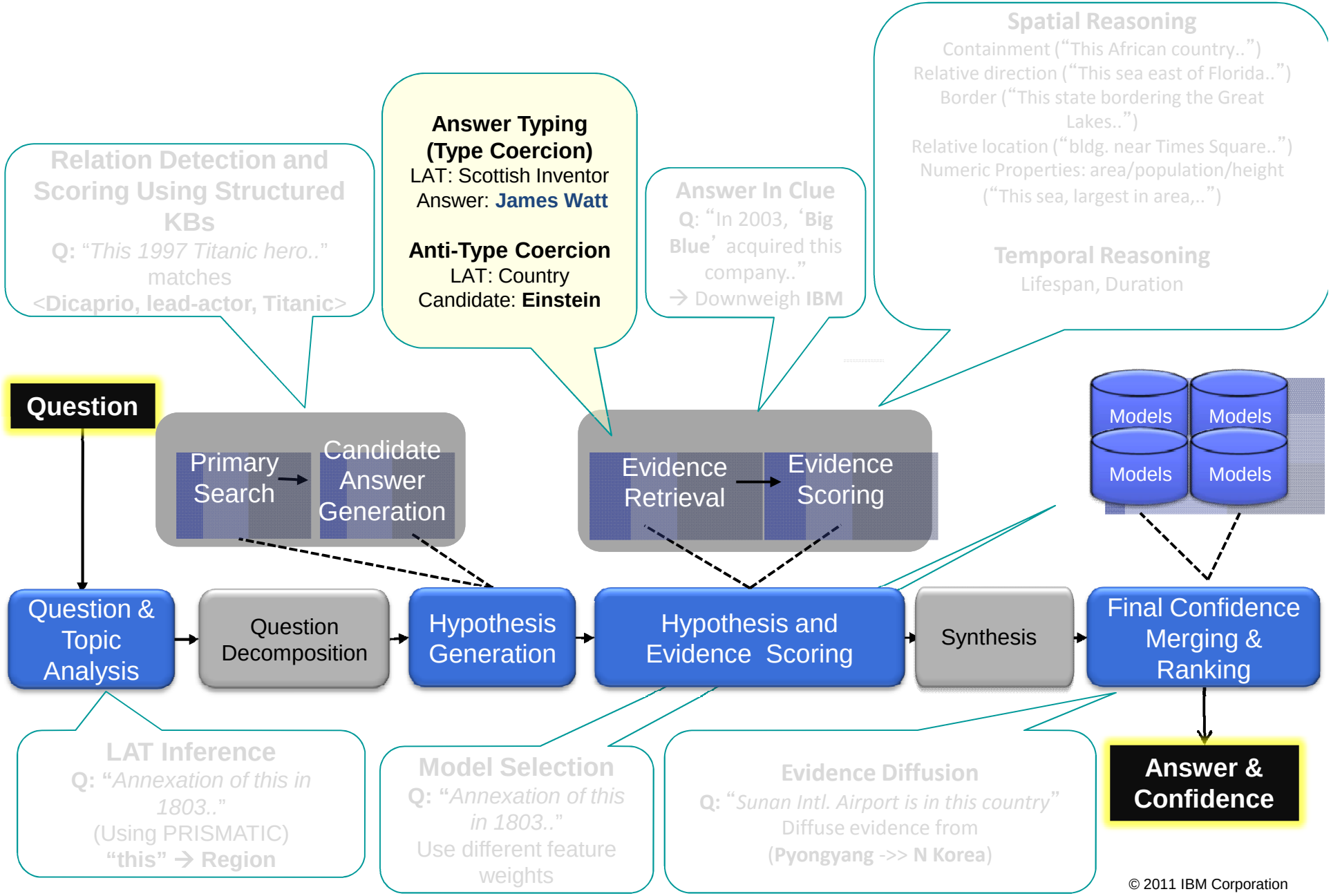
Watson's Architecture



Using Structured Data and Inference

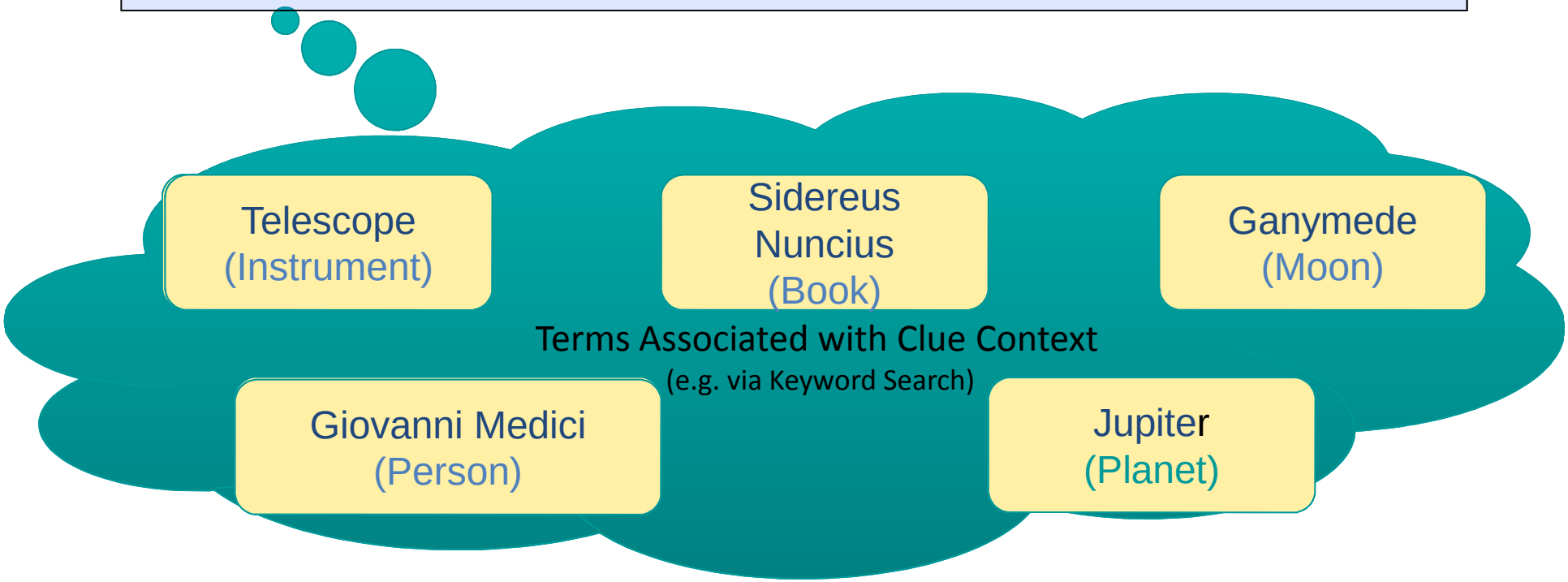


Using Structured Data and Inference



Type Information - a crucial hint to get the correct answer

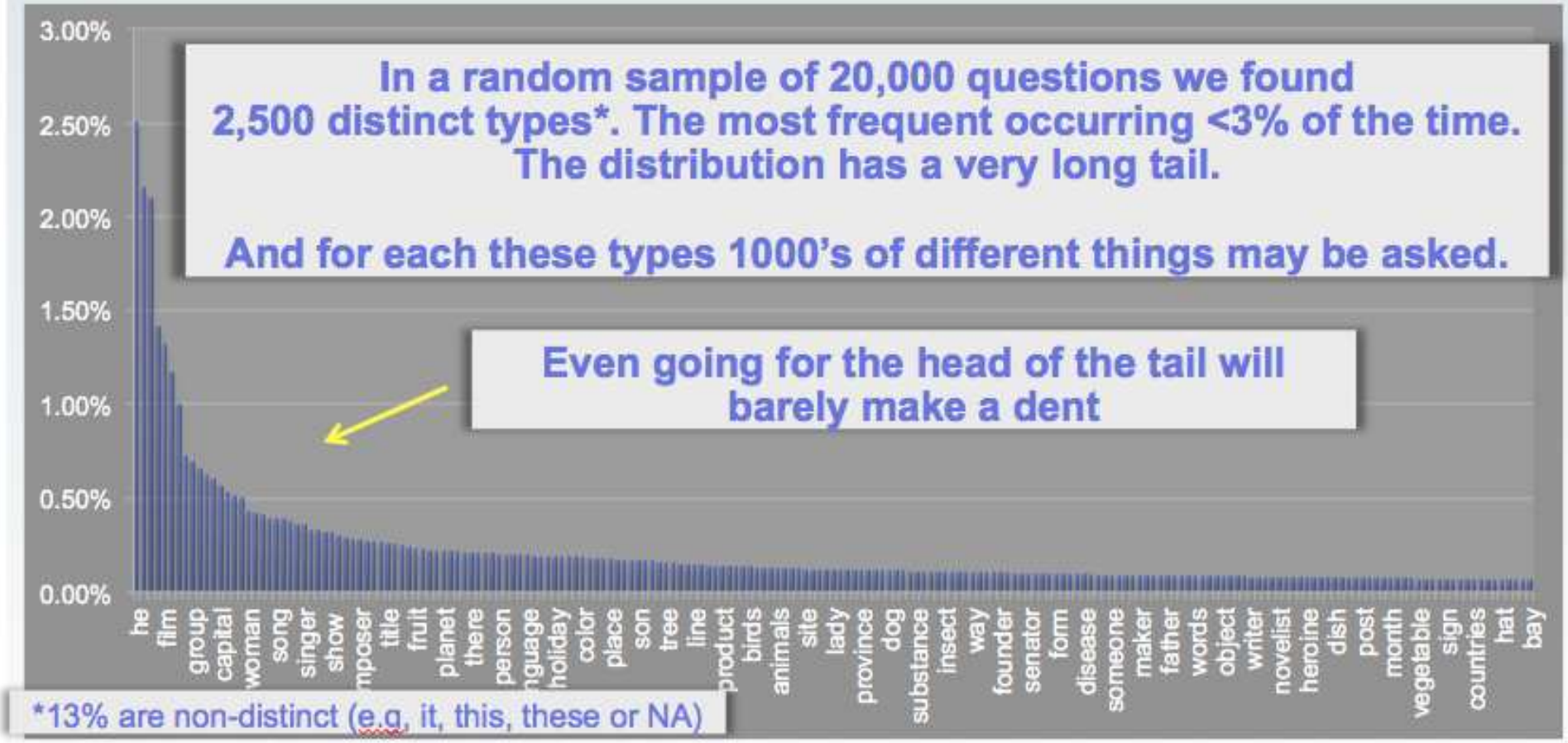
ASTRONOMY: In 1610 Galileo named the moons of this planet for the Medici brothers



Broad Domain

We do NOT attempt to anticipate all questions and build databases.

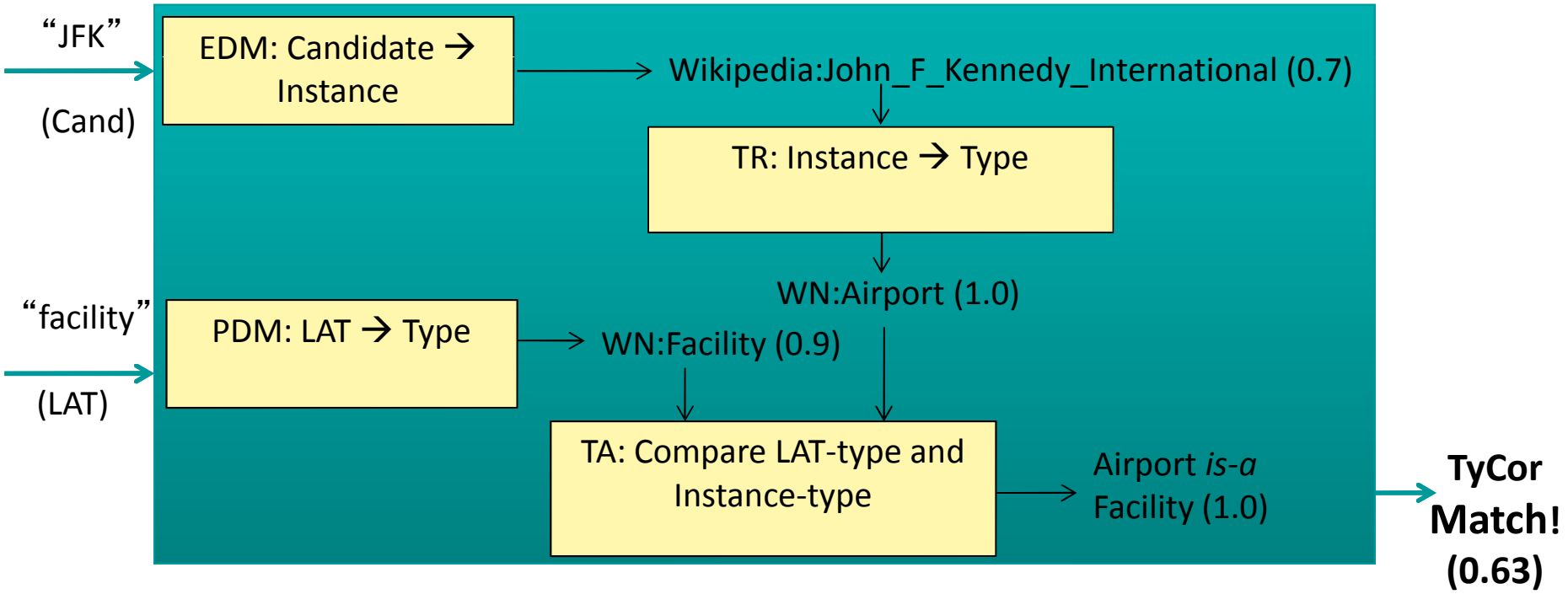
We do NOT try to build a formal model of the world



Our Focus is on reusable NLP technology for analyzing vast volumes of *as-is* text.
Structured sources (DBs and KBs) provide background knowledge for interpreting the text.

TyCor Framework

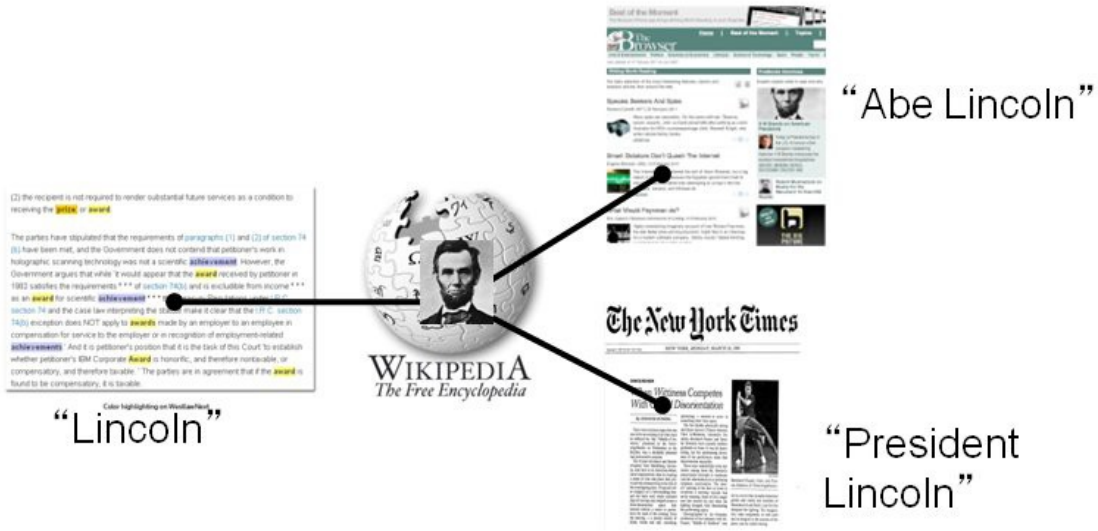
- **Problem**: Compute type match for candidate w.r.t. LAT
  - Both candidate and LAT expressed as Strings
  - **4 Steps: EDM** (Entity Disambiguation and Matching)**, PDM** (Predicate Disambiguation and Matching)**, TR** (Type Retrieval)**, TA** (Type Alignment)



IBM WATSON

Entity Disambiguation and Matching Problem

Fundamental Task in NLP: Map entity string to source identifier (dbpedia URI, UMLS cui, etc.)



Issue 1 (Entity Matching): Many different ways to refer to the same entity (spellings, aliases, nicknames, abbreviations)


World All World Topics

Gaddafi? Kadafi? Qaddafi? What's the correct spelling?

You say, Gaddafi, we say Qaddafi. Other variations on the leader of Libya include "Gathafi," "Kadafi," and "Gadafy," creating an unholy mess for newspaper editors.

Issue 2 (Entity Disambig): Sense Disambiguation depends on context

Flight took off from JFK…	JFK was assassinated…	Film critics loved JFK…

Resources Used In Watson:

For Matching (aka Identification)

- Wikipedia redirects (Myanmar ->> Burma)
- Synonyms / aliases extracted from text

  "*IBM's distinctive culture and product branding has given it the nickname Big Blue*"
- Anchor-Hyperlink Data

For Disambiguation

- Wikipedia Disambiguation Pages (wide coverage)
  - ~150K disambiguation pages in 2008
  - E.g. "Java" has >50 different senses spanning >20 Distinct Types
- Do not need WordNet Synsets for Instances (poor coverage)

EDM Approaches

- Basic EDM Algorithm (2010)
  - **Input**: Entity String
  - **Output**: Set of Wikipedia URIs ranked along following criteria
    1. Exact Match to WP Page Title
    2. Redirect to WP Page
    3. {Disambiguations, Link Anchors, Mined Synonym Lists}

    Entities in (3) sorted by popularity

F1: .83

E.g. "Emerson" → <RW Emerson, 0.7>
<Emerson College, 0.2>
<Emerson Radio, 0.1>

Relevance

- Context-Sensitive EDM (2011)

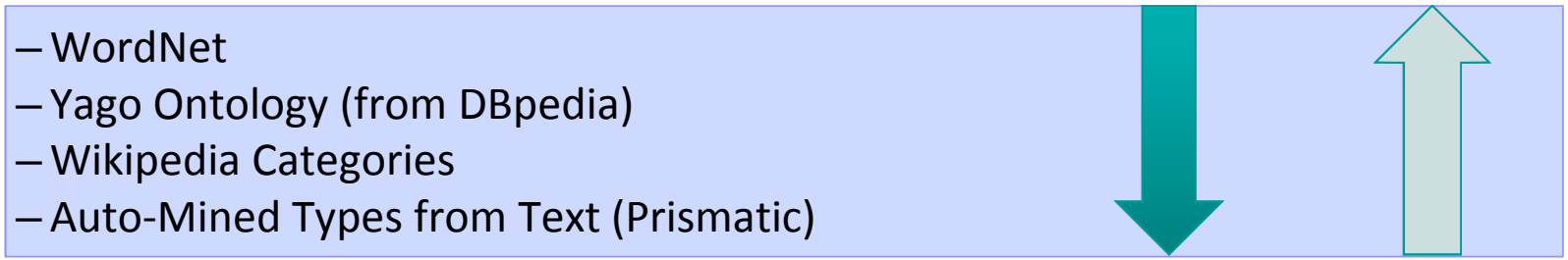
  Use context to disambiguate entities
  - Question text (BoW) as context for question entities, match against:
    - Wikipedia page text for dbPedia entities (BoW)
    - Label, variant, and relational neighbors for Freebase, GeoNames
  - Look at relations involving entity
    ("JFK's brother Teddy" -> JFK hasBrother TeddyKennedy)
  - Train & Evaluate against wikiPedia link-anchor text

F1: .89

Note: component level improvements reach a effort/effect saturation. This did not improve Jeopardy! performance and was resource-intensive – not used in J! Watson

- Obtain Types for Instances returned by EDM
- Taxonomies Used In Watson:

RECALL PRECISION



- Interesting Points
  - Type Systems are linked
    - Yago → WordNet
  - Wiki-Categories contain extra information (modifiers)
    - Einstein : German-Inventor, Swiss-Vegetarian, Patent-Examiner
  - Automatically Mined Types reflect real world usage
    - Fluid -is-a- Liquid (strictly speaking incorrect)

PDM

IBM WATSON

- Predicate Disambiguation and Matching (PDM) Problem (basically WSD)
  - LAT: star

In the northern hemisphere, latitude is equal to the angle above the horizon of this star, Alpha Ursae Minoris	***This star of "The Practice" played Clint Eastwood's Secret Service partner in the film "In the Line of Fire"***

- Similar in principle to EDM
  - EDM – map named entity → instance
  - PDM – map generic noun → class/type

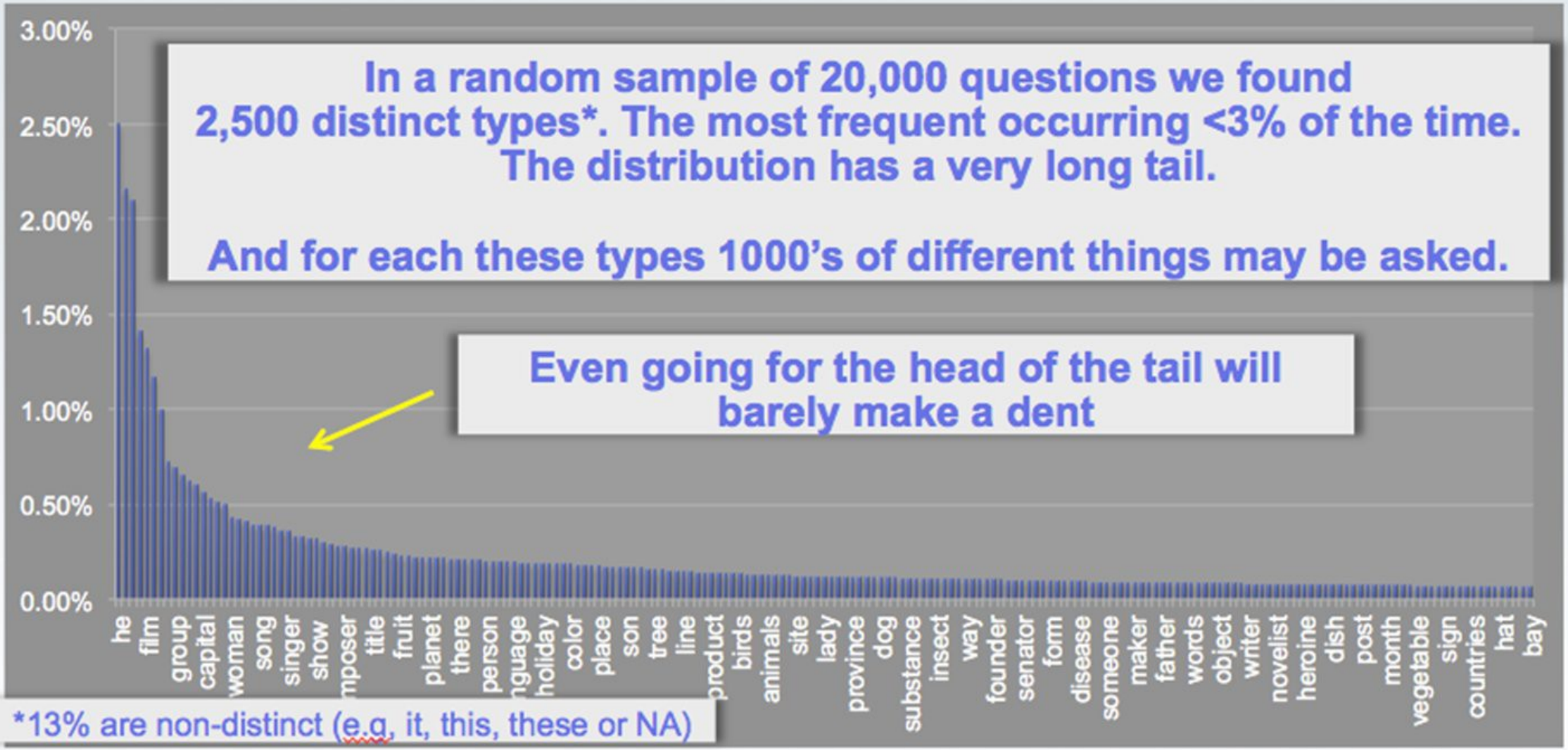
- PDM in Watson:
  - Lookup LAT in WordNet: Order concepts based on sense ranks
  - Special Case: Domain-specific PDM (manually specified)
    - Mapped top-200 LATs in Jeopardy! to specific concept senses

Broad Domain

IBM WATSON

We do NOT attempt to anticipate all questions and build databases.

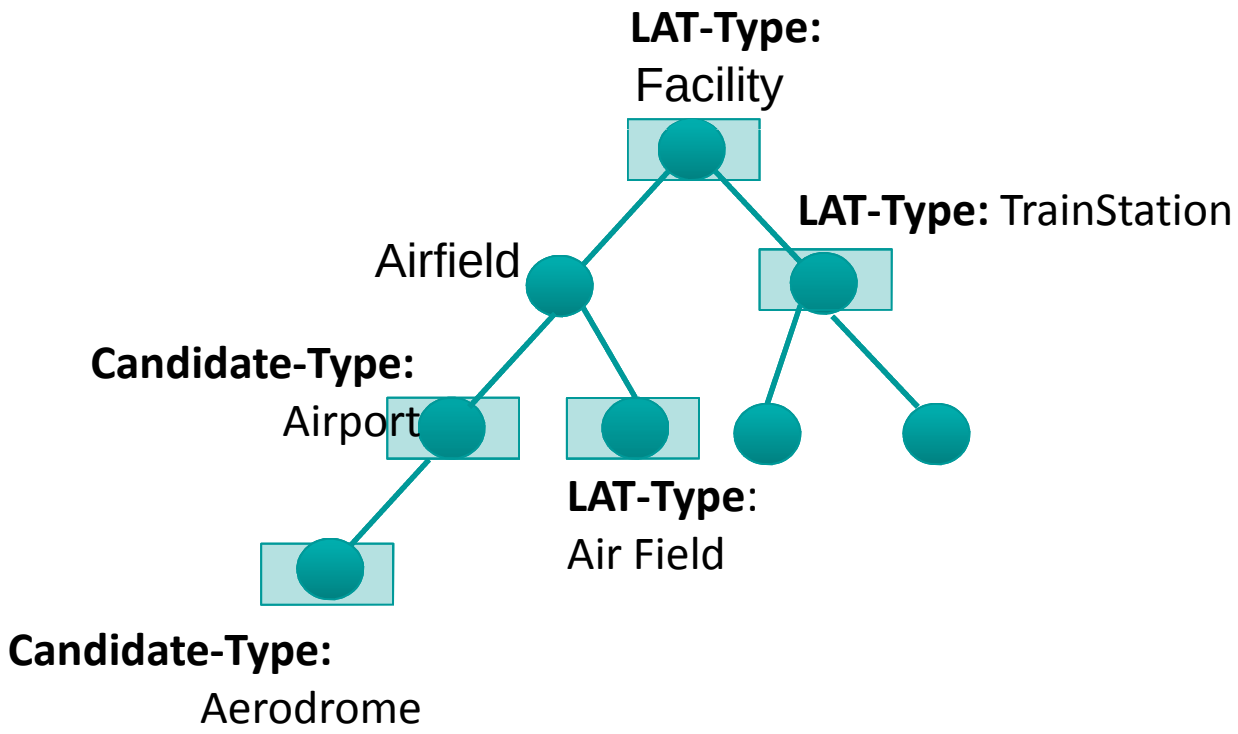
We do NOT try to build a formal model of the world



Our Focus is on reusable NLP technology for analyzing vast volumes of *as-is* text.

Structured sources (DBs and KBs) provide background knowledge for interpreting the text.

- Type Matching/Alignment Problem
  - Compare candidate types with LAT types
  - Produce a score depending degree of Match
- Various Types of Match Considered



Subclass Match (1.0)

Sibling Match (0.5)

Deep LCA Match (0.25)

Disjoint Types (-1.0)

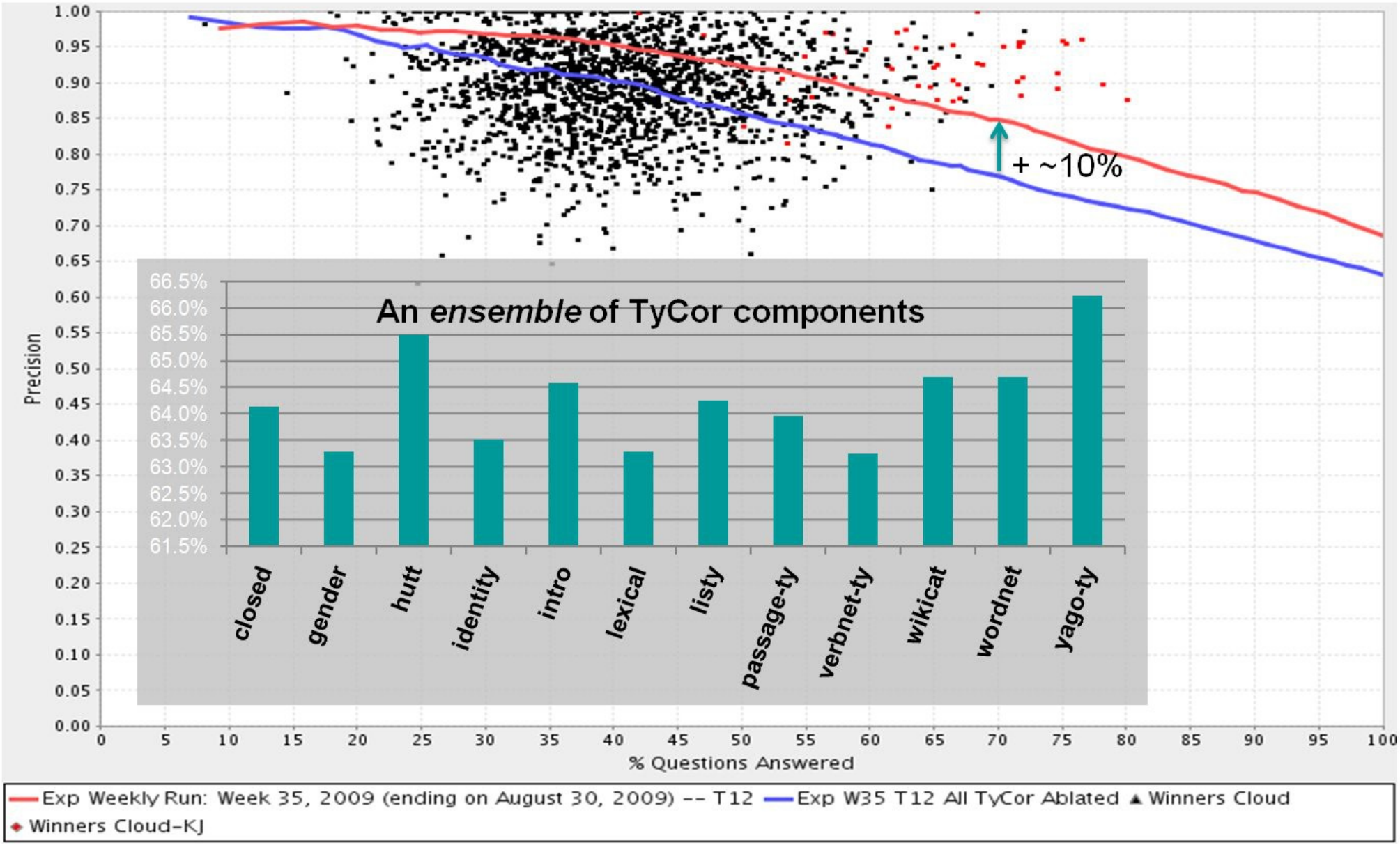
Putting it all together

TyCor Score = EDM * TR * PDM * TA

- Intermediate Failure
  - If any step fails, Tycor Score = 0 (consider smoothing)
  - Expose which step failed to final model (EDM-Failure, PDM-Failure…)
- An-TyCor
  - When TA score is -1 (Disjoint Types) → AnTyCor Feature added to model
  - Strong negative signal against candidate
  - Helps rules out candidates of wrong type (e.g. LAT: Country, Candidate: Einstein)
- Multiple LATs
  - When multiple LATs in question with confidences: (L1, L2..Ln)
  - Final TyCor Score (weighted-sum) = (L1 * Tyc1) + (L2 * Tyc2) + .. (Ln * Tycn)

- TyCor Algorithm Suite in J! Watson
  - -14 TyCors Developed
  - All TyCors follow 4 steps
  - Each TyCor score is a separate feature in model
  - Model learns weights on diff. TyCors: balances/combines type information
  - Trained on QA ground truth, not type ground truth

TyCor Impact on *Jeopardy!* clues



Coverage Issues

- Knowledge-bases (e.g. dbPeda, Freebase) are imperfect
  - 80% of Jeopardy answers were in dbPedia
  - 15% of the name variants were missing
  - Errors found in 10% of entries (names, improper classification, incorrect relations)
  - Typing relations useful in 85% of *Jeopardy!* questions
  - all other relations combined to impact fewer than 15% of questions
- The fundamental barrier:
  - Given a knowledge graph triple

    P(triple is useful) = P(X is mentioned in a q) * P(one of Xs known variants was used in q) * P(R is useful information for answering q) * P(Y is the answer or leads to the answer) * P(one of Ys known variants is used in answer evidence)

  - Given a q about some X "*What nation borders S. Korea*"

    P(there is a useful triple) = P(X is in the graph) * P(X variant used in q is known) * P(the q requires a relation R that is in the graph) * P(R can be recognized in q) * P(X is connected by R to Y) * P(Y is useful for answering q) * P(one of Ys known variants is used in answer evidence
- We did not *fix* any sources, we needed accurate models of impact on *Jeopardy*!
- EDM is the gateway to exploiting structured sources
- For Medical QA, EDM is a major bottleneck