<u>Watson</u>

Question Answering (QA)

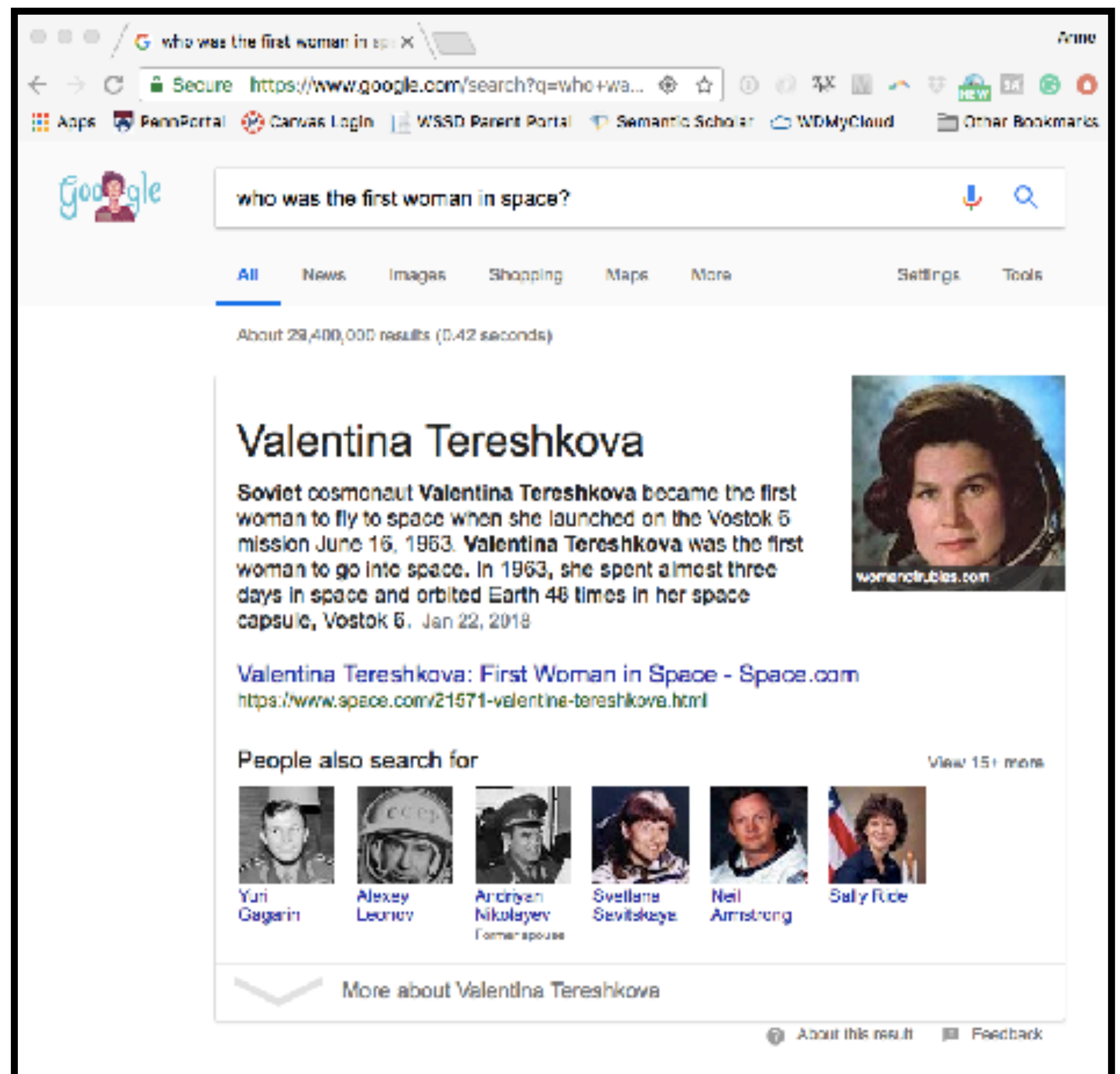
CIS 530

March 26, 2018

Jurafsky & Martin Ch. 28 (draft):
https://web.stanford.edu/~jurafsky/slp3/28.pdf


WolframAlpha
computational knowledge engine.
Enter what you want to calculate or know about:
Web Apps
Examples
Random


AT&T
12:05 PM
Who won the Sixers game last night
Tap to Edit
The 76ers overcame the Grizzlies by a score of 119 to 105 yesterday.
SPORTS
105-119
Grizzlies
Final - Yesterday
76ers
1 2 3 4 T
Grizzlies 19 25 25 36 105
76ers 26 32 41 20 119
Wells Fargo Center
See More on Yahoo
YAHOO!


Secure https://www.google.com/search?q=who+wa...
who was the first woman in space?
All News Images Shopping Maps More Settings Tools
About 29,400,000 results (0.42 seconds)
Valentina Tereshkova
Soviet cosmonaut Valentina Tereshkova became the first woman to fly to space when she launched on the Vostok 6 mission June 16, 1963. Valentina Tereshkova was the first woman to go into space. In 1963, she spent almost three days in space and orbited Earth 48 times in her space capsule, Vostok 6. Jan 22, 2018
Valentina Tereshkova: First Woman in Space - Space.com
https://www.space.com/21571-valentina-tereshkova.html
People also search for
Yuri Gagarin
Alexey Leonov
Andriyan Nikolayev
Svetlana Savitskaya
Neil Armstrong
Sally Ride
More about Valentina Tereshkova
Feedback

Today’s Objectives

- Understand the basics of factoid QA systems:
    - Information Retrieval-based
    - Knowledge-based
- Be able to explain the major challenges associated with automating QA
- Familiarity with recent research directions (deep learning)

Much of the work in QA focuses on factoid questions

`Q: Which US state capital has the largest population?`

Much of the work in QA focuses on factoid questions

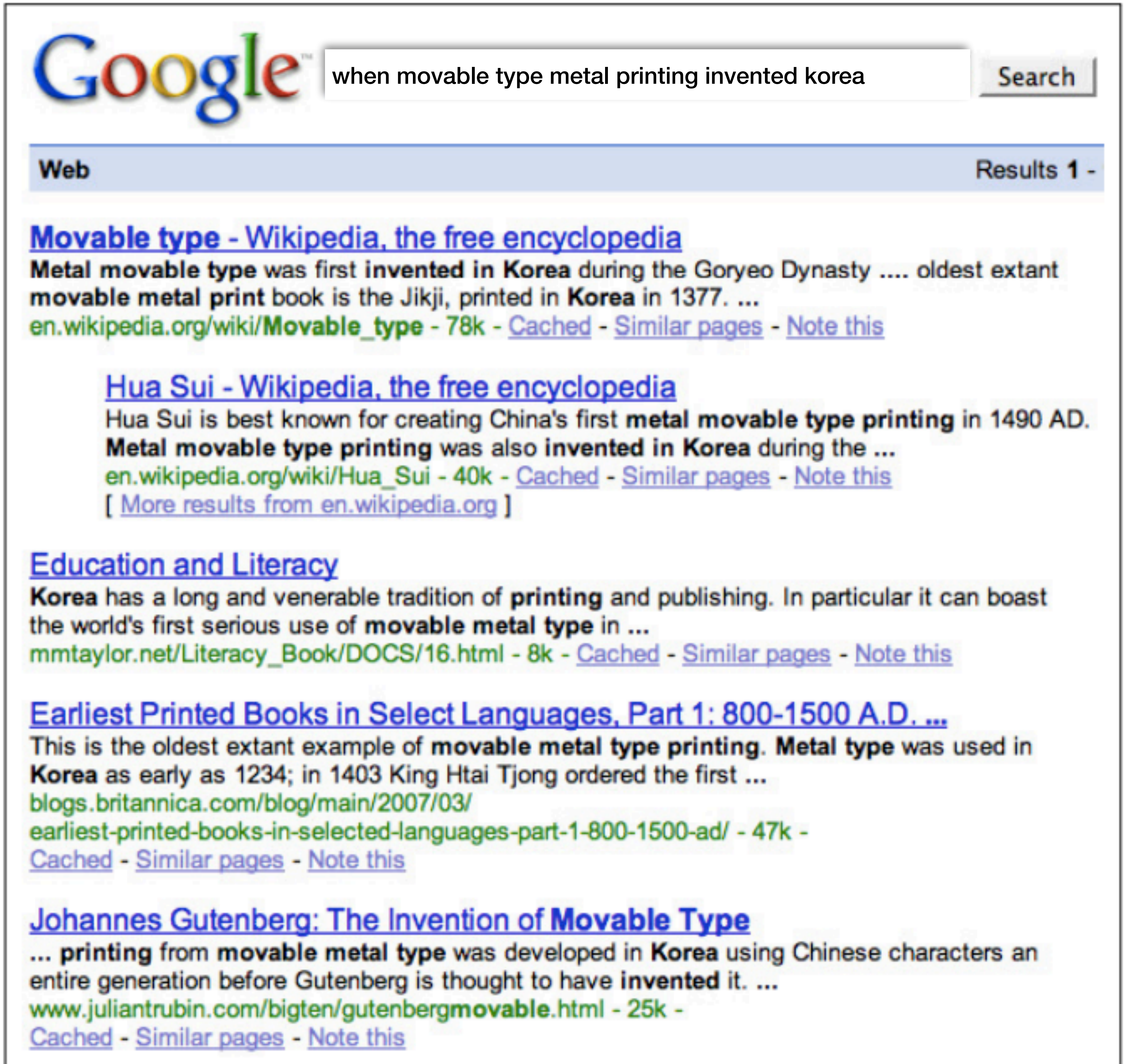
`Q: Which US state capital has the largest population?`

`A: Phoenix, AZ`

Two major paradigms for QA methods

- Information Retrieval (IR)-based question answering
- Knowledge-based question answering

When was movable type metal printing invented in Korea?



when movable type metal printing invented korea

Search

Web Results **1** -

[Movable type - Wikipedia, the free encyclopedia]
Metal movable type was first **invented in Korea** during the Goryeo Dynasty .... oldest extant **movable metal print** book is the Jikji, printed in **Korea** in 1377. ...
en.wikipedia.org/wiki/**Movable_type** - 78k - Cached - Similar pages - Note this

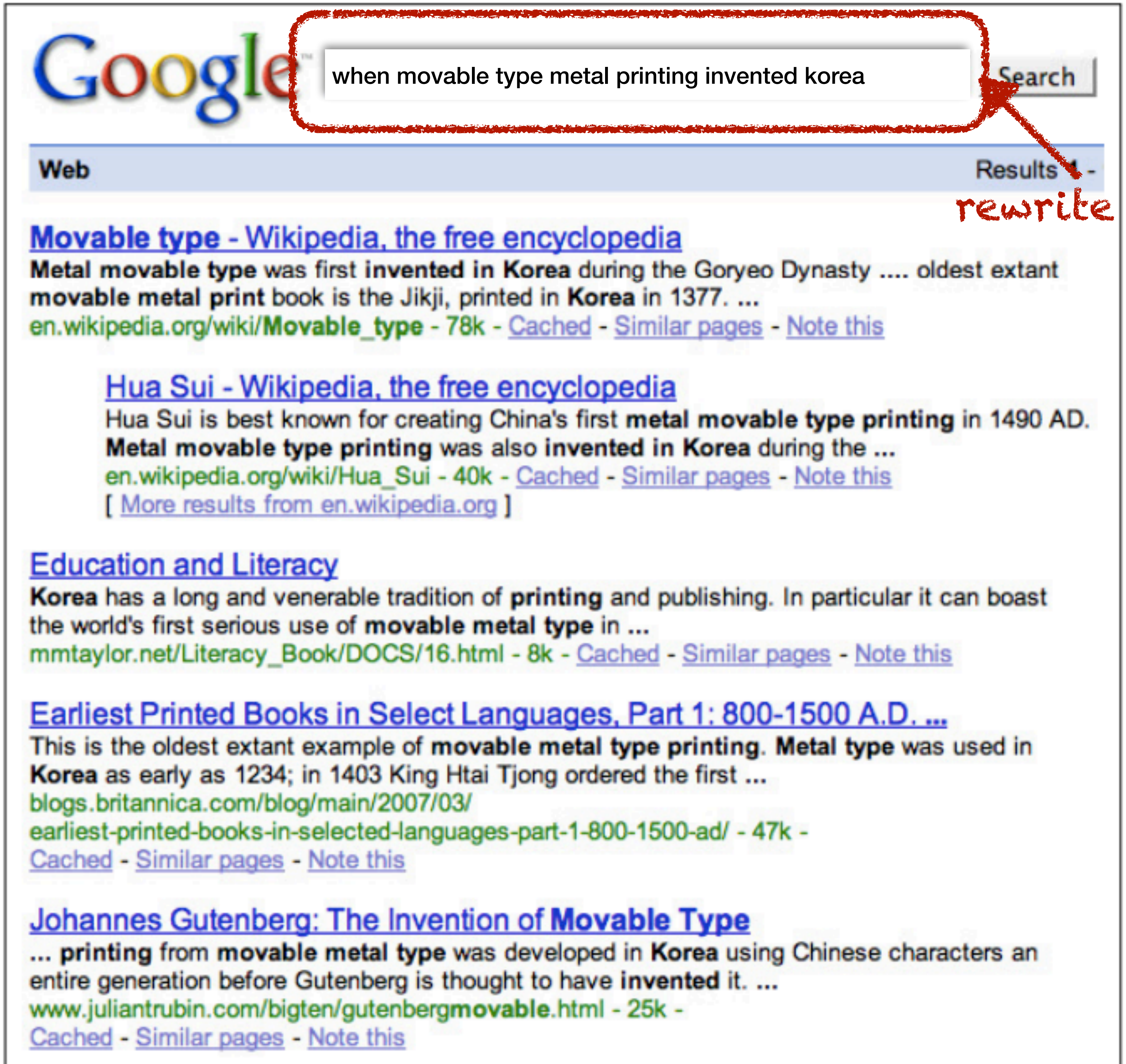
[Hua Sui - Wikipedia, the free encyclopedia]
Hua Sui is best known for creating China's first **metal movable type printing** in 1490 AD. **Metal movable type printing** was also **invented in Korea** during the ...
en.wikipedia.org/wiki/Hua_Sui - 40k - Cached - Similar pages - Note this
[More results from en.wikipedia.org]

[Education and Literacy]
Korea has a long and venerable tradition of **printing** and publishing. In particular it can boast the world's first serious use of **movable metal type** in ...
mmtaylor.net/Literacy_Book/DOCS/16.html - 8k - Cached - Similar pages - Note this

[Earliest Printed Books in Select Languages, Part 1: 800-1500 A.D. ...]
This is the oldest extant example of **movable metal type printing**. **Metal type** was used in **Korea** as early as 1234; in 1403 King Htai Tjong ordered the first ...
blogs.britannica.com/blog/main/2007/03/
earliest-printed-books-in-selected-languages-part-1-800-1500-ad/ - 47k -
Cached - Similar pages - Note this

[Johannes Gutenberg: The Invention of **Movable Type**]
... **printing** from **movable metal type** was developed in **Korea** using Chinese characters an entire generation before Gutenberg is thought to have **invented** it. ...
www.juliantrubin.com/bigten/gutenberg**movable**.html - 25k -
Cached - Similar pages - Note this

When was movable type metal printing invented in Korea?


Google

when movable type metal printing invented korea

Search

Web

Results 1 -

[Movable type - Wikipedia, the free encyclopedia]
Metal movable type was first **invented in Korea** during the Goryeo Dynasty .... oldest extant **movable metal print** book is the Jikji, printed in **Korea** in 1377. ...
en.wikipedia.org/wiki/**Movable_type** - 78k - Cached - Similar pages - Note this

[Hua Sui - Wikipedia, the free encyclopedia]
Hua Sui is best known for creating China's first **metal movable type printing** in 1490 AD. **Metal movable type printing** was also **invented in Korea** during the ...
en.wikipedia.org/wiki/Hua_Sui - 40k - Cached - Similar pages - Note this
[More results from en.wikipedia.org]

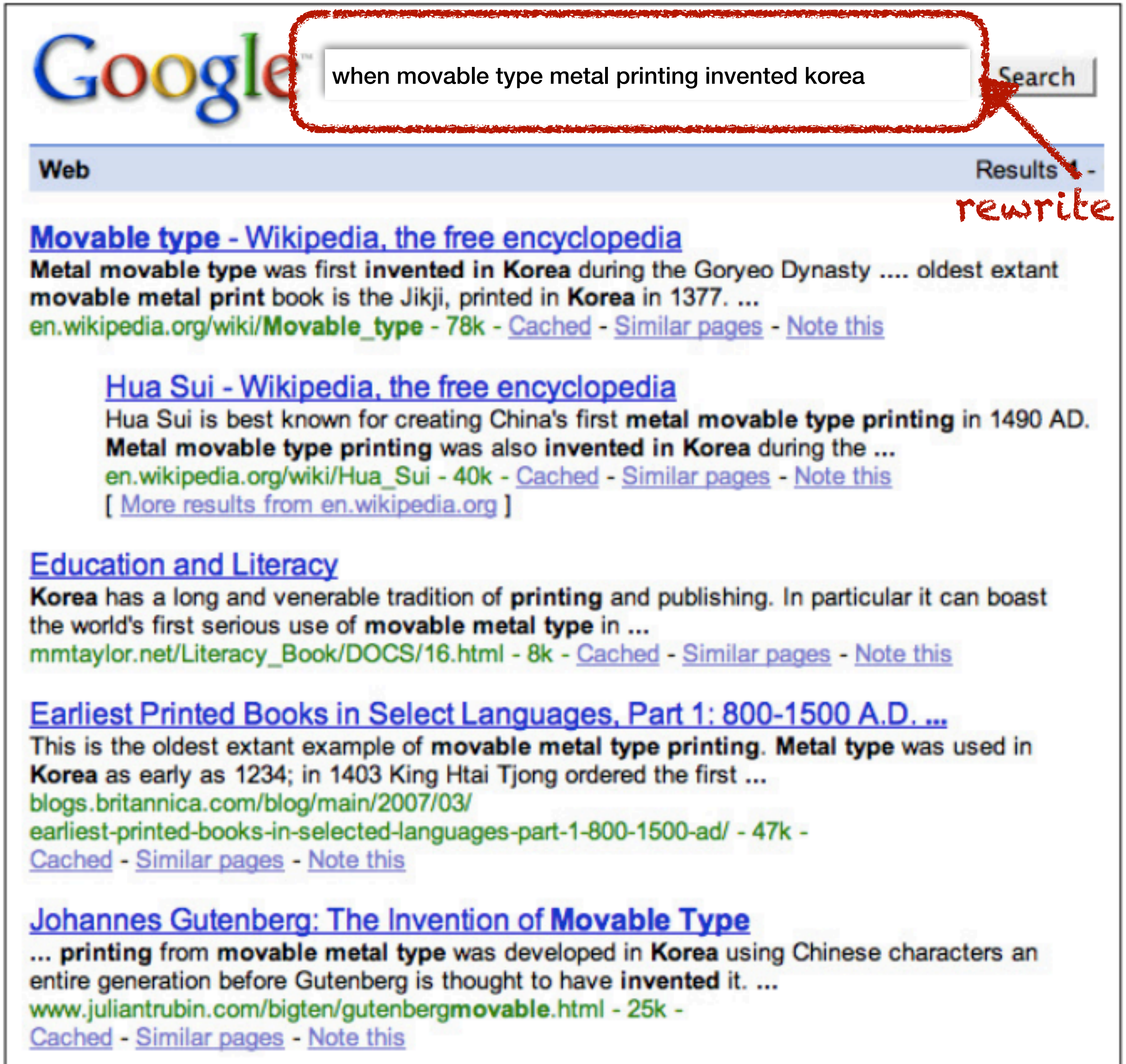
[Education and Literacy]
Korea has a long and venerable tradition of **printing** and publishing. In particular it can boast the world's first serious use of **movable metal type** in ...
mmtaylor.net/Literacy_Book/DOCS/16.html - 8k - Cached - Similar pages - Note this

[Earliest Printed Books in Select Languages, Part 1: 800-1500 A.D. ...]
This is the oldest extant example of **movable metal type printing**. **Metal type** was used in **Korea** as early as 1234; in 1403 King Htai Tjong ordered the first ...
blogs.britannica.com/blog/main/2007/03/
earliest-printed-books-in-selected-languages-part-1-800-1500-ad/ - 47k -
Cached - Similar pages - Note this

[Johannes Gutenberg: The Invention of **Movable Type**]
... **printing** from **movable metal type** was developed in **Korea** using Chinese characters an entire generation before Gutenberg is thought to have **invented** it. ...
www.juliantrubin.com/bigten/gutenberg**movable**.html - 25k -
Cached - Similar pages - Note this

rewrite the query

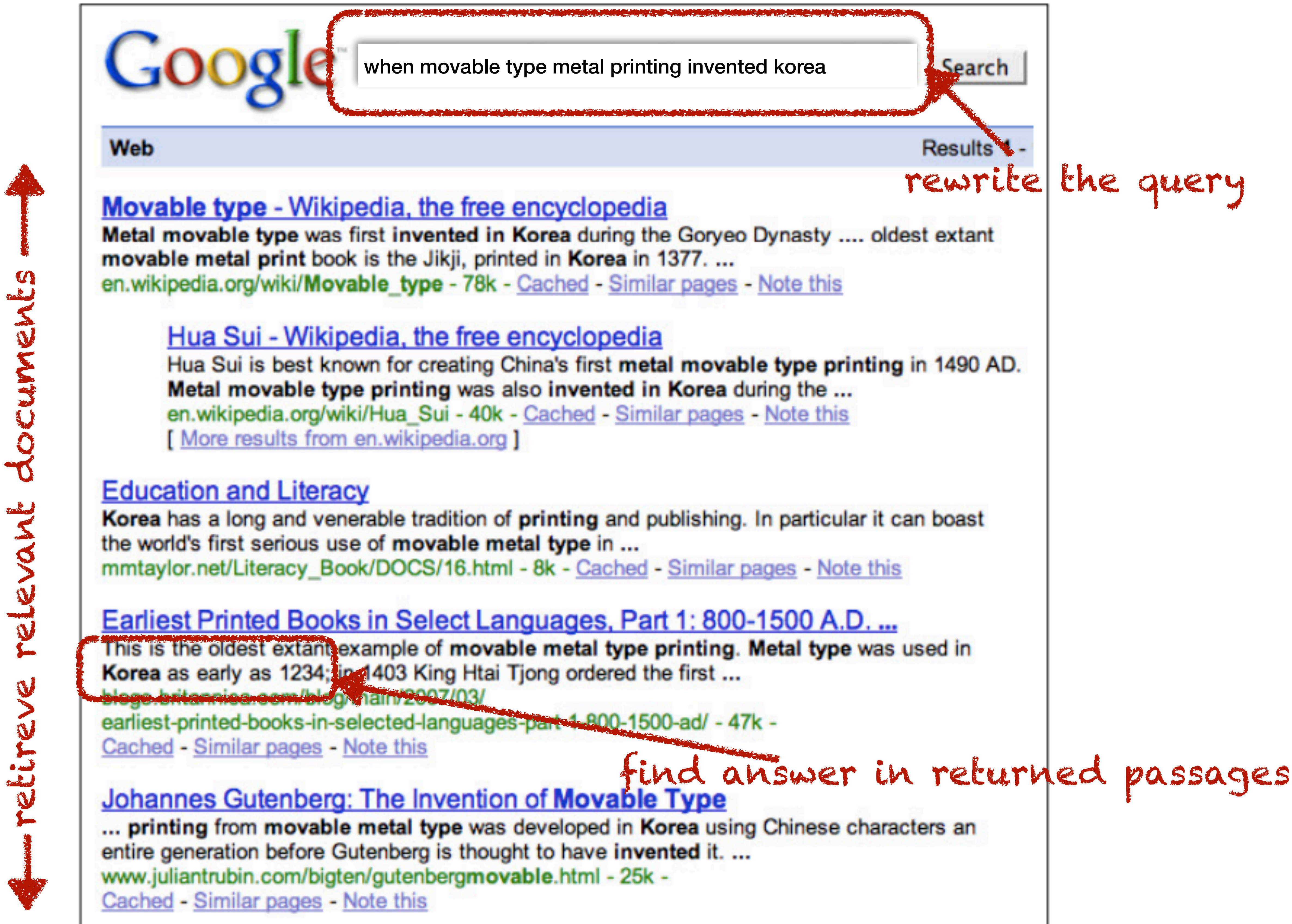
When was movable type metal printing invented in Korea?



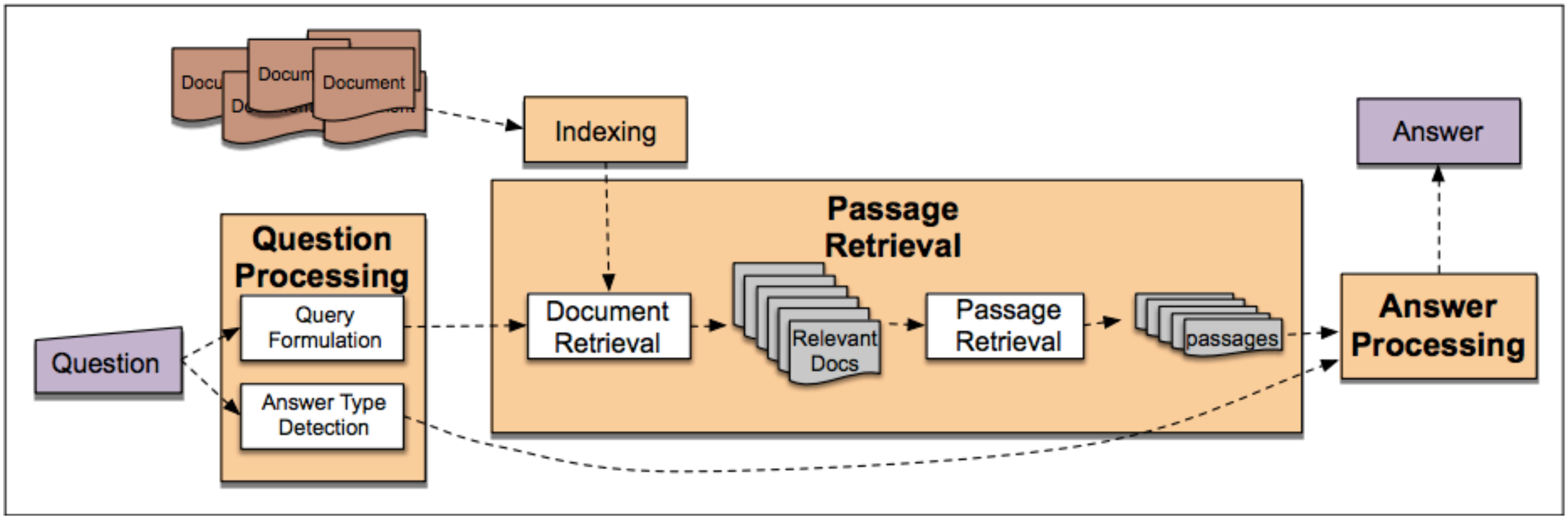
rewrite the query

retireve relevant documents

When was movable type metal printing invented in Korea?



The three stages of IR-based QA



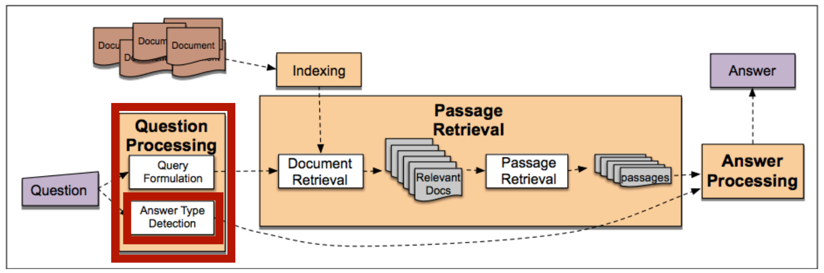
The <u>question processing</u> step formulates a query and predicts the answer type

Given:

Q: Which US state capital has the largest population?

The query processing step should produce results like this:

Answer Type	city
Query	US state capital, largest, population
Focus	state capital

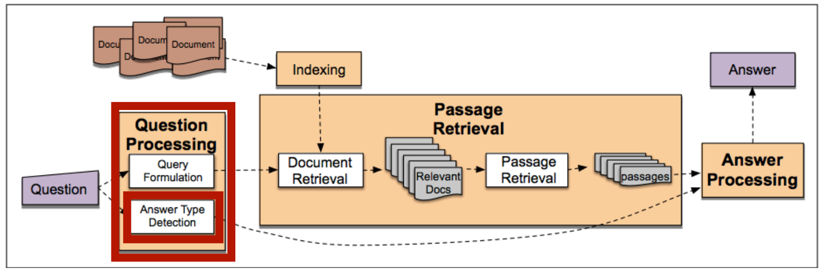


Predicting the <u>answer type</u> enables us to focus our search

Goal of answer type detection:

Given a question, predict the *type* of answer we're looking for

Question	Answer Type
Who founded Virgin Airlines?	PERSON
Which Canadian city has the largest population?	CITY

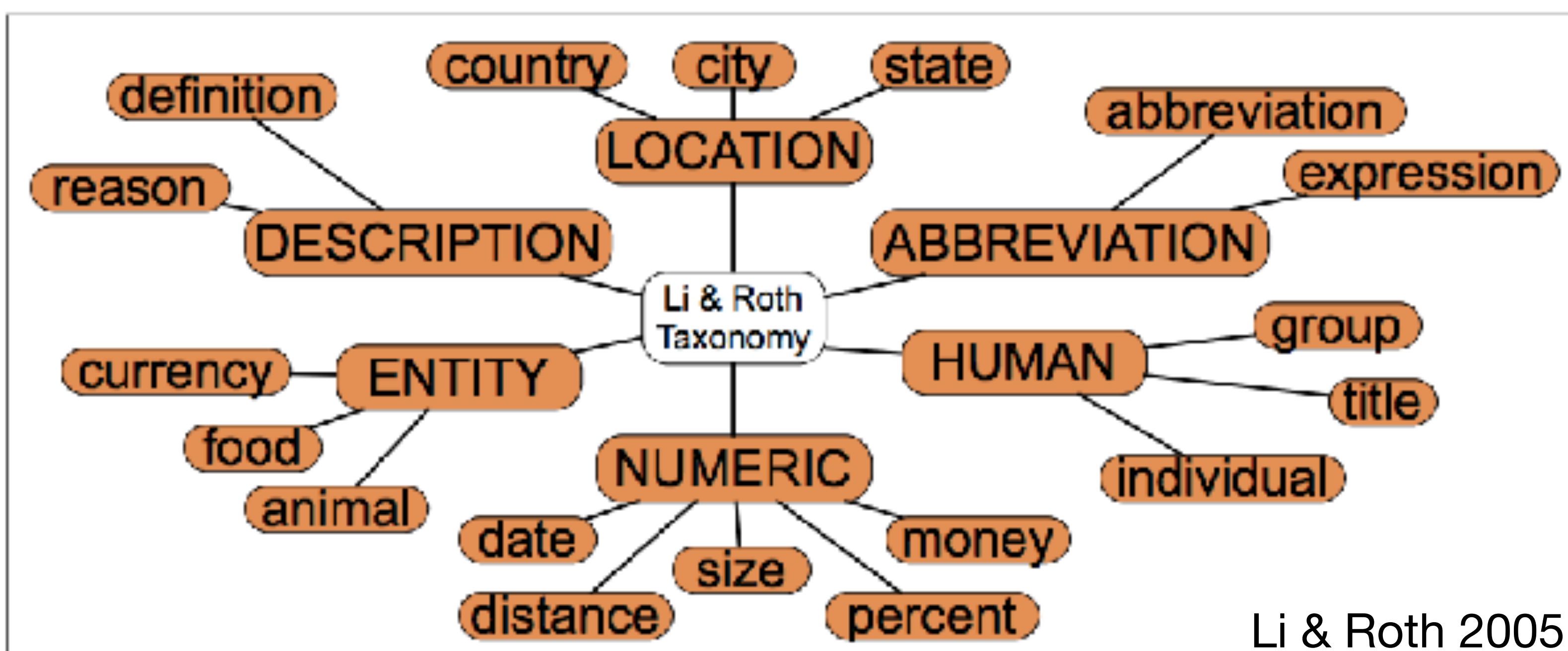


Predicting the <u>answer type</u> enables us to focus our search

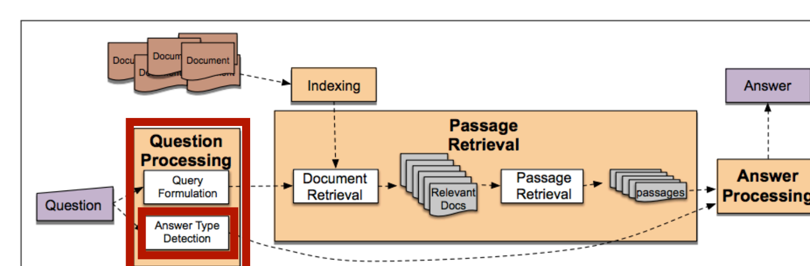
Goal of answer type detection:
Given a question, predict the *type* of answer we're looking for

Question	Answer Type
Who founded Virgin Airlines?	PERSON
Which Canadian city has the largest population?	CITY

Answer types come from a pre-defined set:



Li & Roth 2005



Question:

What types of features would you use to predict answer type?

(Chat with the person next to you, 2 mins)

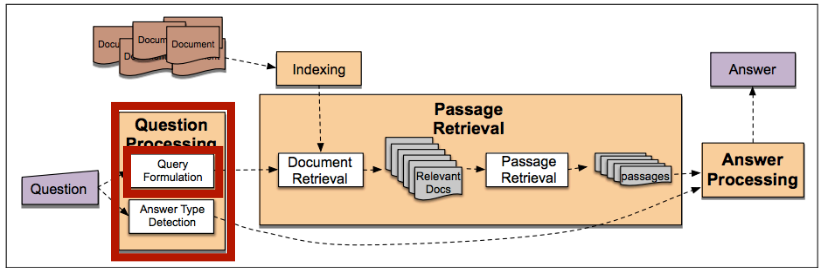
Question	Answer Type
Who founded Virgin Airlines?	PERSON
Which Canadian city has the largest population?	CITY

<u>Query formulation</u> generates the query to pass to the IR system

Goal of query formulation:

Re-format the question into input for the IR system

`Q: Who founded Virgin Airlines?` → `QUERY: “Virgin Airlines founder”`



Query formulation generates the query to pass to the IR system

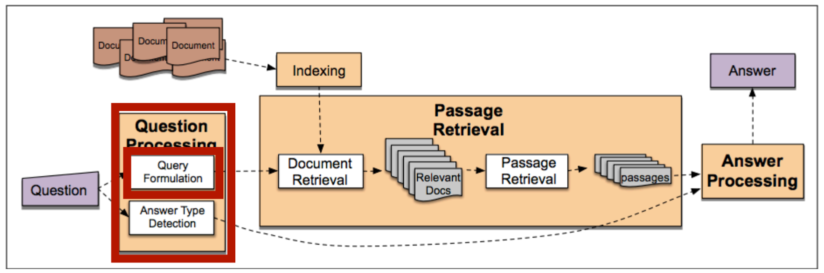
Goal of query formulation:
Re-format the question into input for the IR system

`Q: Who founded Virgin Airlines?` → **`QUERY: “Virgin Airlines founder”`**

- **Query Re-formulation:**
  - Re-phrase the question to make it look like a substring of possible answers

`Where is A –> A located in`

`Why did A <VERB> B –> A <VERB+ED> B because`



Query formulation generates the query to pass to the IR system

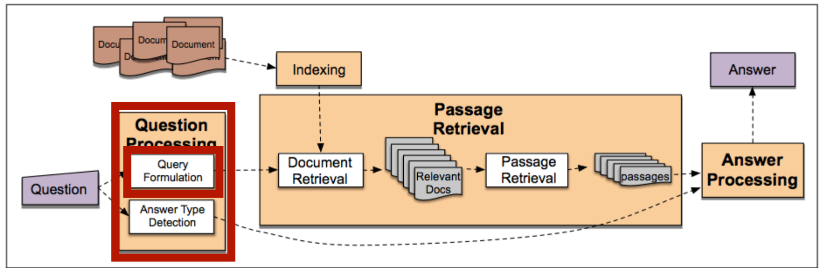
Goal of query formulation:

Re-format the question into input for the IR system

`Q: Who founded Virgin Airlines?` → `QUERY: “Virgin Airlines founder”`

- **Query Expansion:**
  - Add additional keywords that might be used in the answer

`Q: Who founded Virgin Airlines?`
`QUERY: “Virgin Airlines founder creator started launched”`

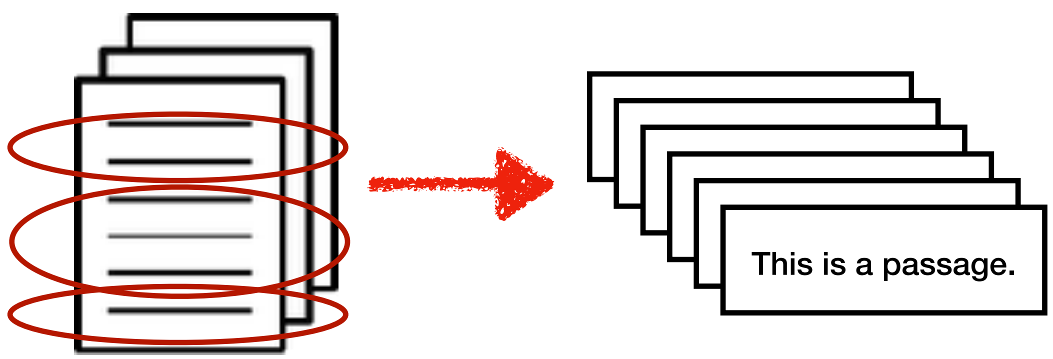


<u>Passage retrieval</u> finds relevant passages in documents retrieved by an IR system

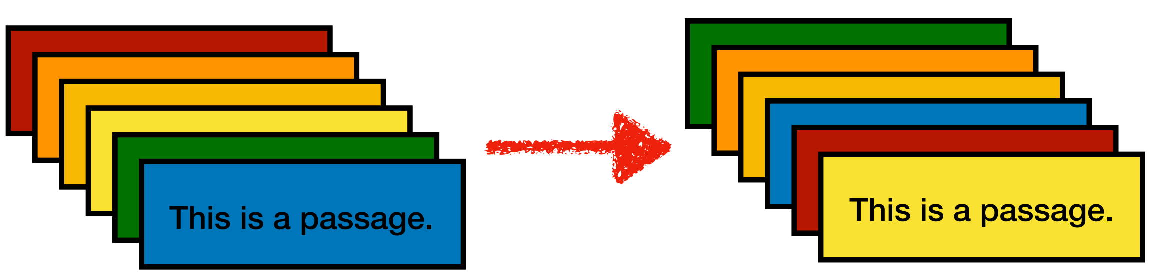
Steps to passage retrieval:

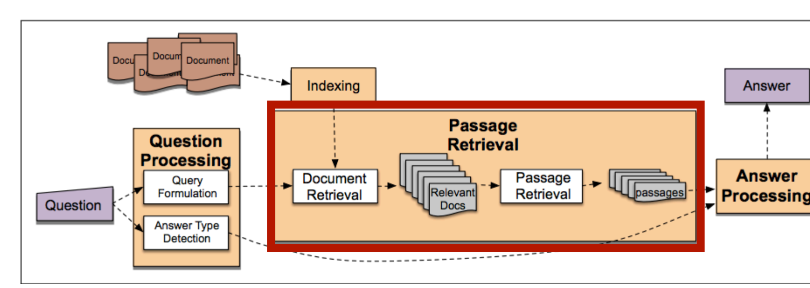
- Pass the reformulated query to a document IR system

- Break the retrieved documents into passages



- Rank the passages for relevance



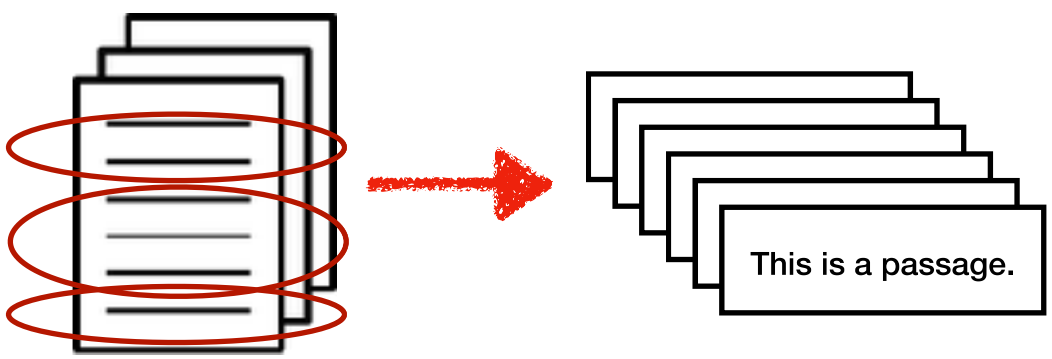


<u>Passage retrieval</u> finds relevant passages in documents retrieved by an IR system

Steps to passage retrieval:

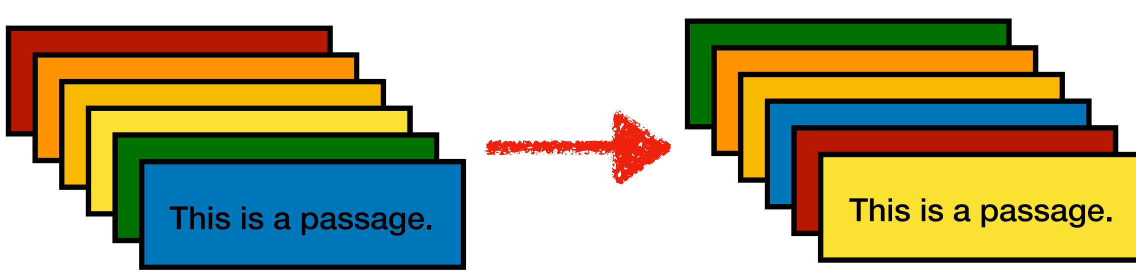
- Pass the reformulated query to a document IR system

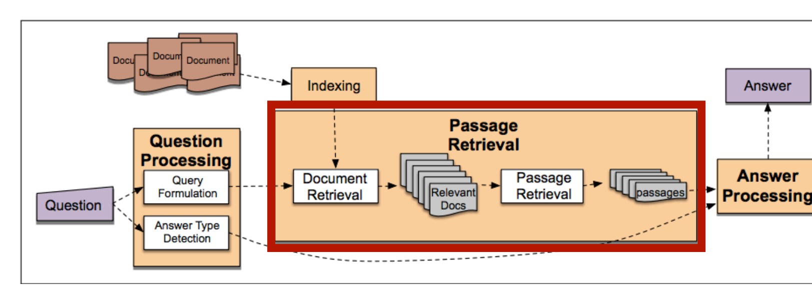
- Break the retrieved documents into passages



Features?

- Rank the passages for relevance



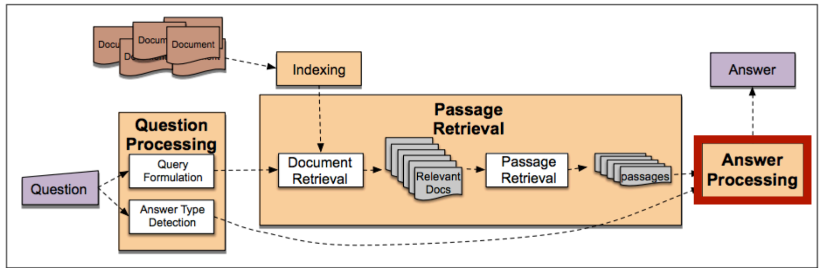


<u>Answer processing</u> extracts a specific answer from the matched passage

Question	**Answer Type**	**Passage**
Who is the prime minister of India?	PERSON	<u>**Narendra Modi**</u>, Prime Minister of India, had told left leaders that the deal would be renegotiated
How tall is Mt. Everest?	NUMERIC: SIZE	The official height of Mt. Everest is <u>**29029 feet**</u>.

Some passages are simple:

Extract the entity with the specified Answer Type



<u>Answer processing</u> extracts a specific answer from the matched passage

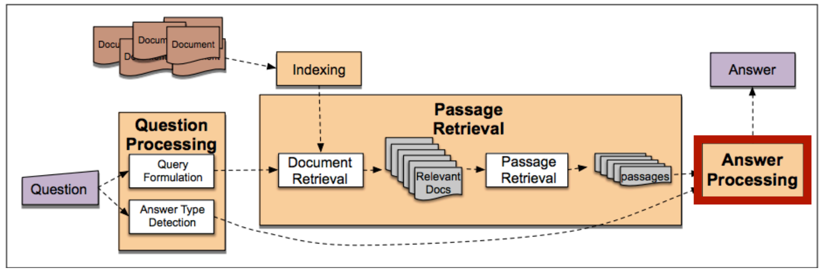
Question	Answer Type	Passage
What is a caldera?	DESCRIPTION: DEFINITION	She made the discovery in the Long Valley caldera, a **<u>volcanic crater</u>** 19 miles long.

<QUESTION PHRASE>, a <ANSWER PHRASE> …

caldera, a volcanic crater …

Some Answer Types are more challenging:

One technique is to apply pattern matching; we might expect a DEFINITION answer to be of the form "<QUESTION PHRASE>, a <ANSWER PHRASE>"



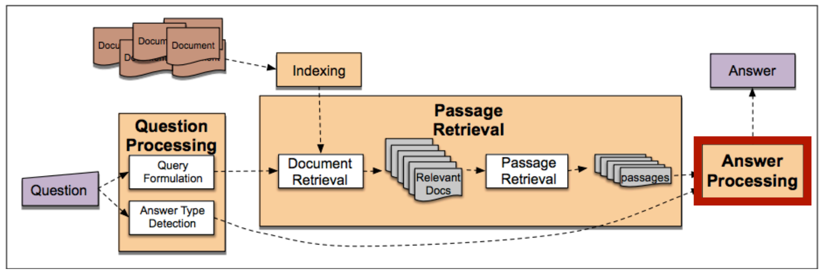
<u>Answer processing</u> extracts a specific answer from the matched passage

Question	**Answer Type**	**Passage**
Who was Queen Victoria's second son?	PERSON	The Marie biscuit is named after **Marie Alexandrovna**, the daughter of **Czar Alexander II of Russia** and wife of **Alfred**, the second son of **Queen Victoria** and **Prince Albert**.

What do we do when there are multiple entities of the specified type?

Use machine learning: Extract features to rank candidate answers

- Answer type match?
- Pattern match?
- Question keywords
- Keyword distance
- Novelty factor
- Apposition features …



Two major paradigms for QA methods

- Information Retrieval (IR)-based question answering
- Knowledge-based question answering

<u>Knowledge-based</u> QA maps questions to queries over a structured database

Take a plain-text question:

```
Q: When was Ada Lovelace born?
```

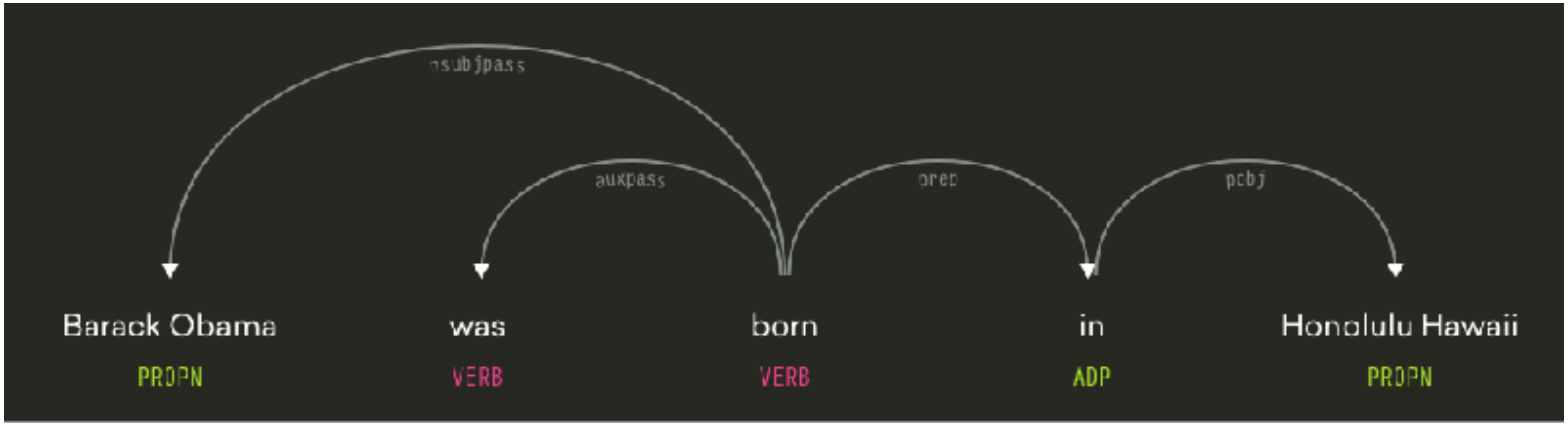
Map it to some query language:

```
BirthYear.AdaLovelace
```

Apply the query to an underlying database:

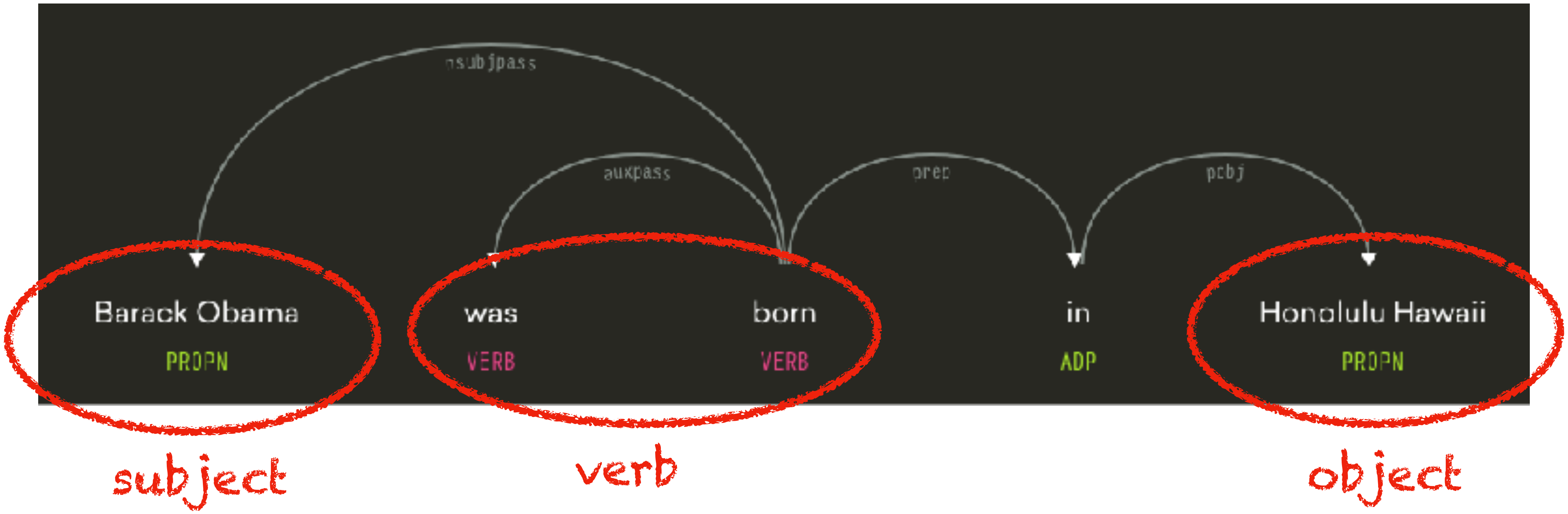
Refresher: Subject-Verb-Object Triples

Barack Obama was born in Honolulu, Hawaii.



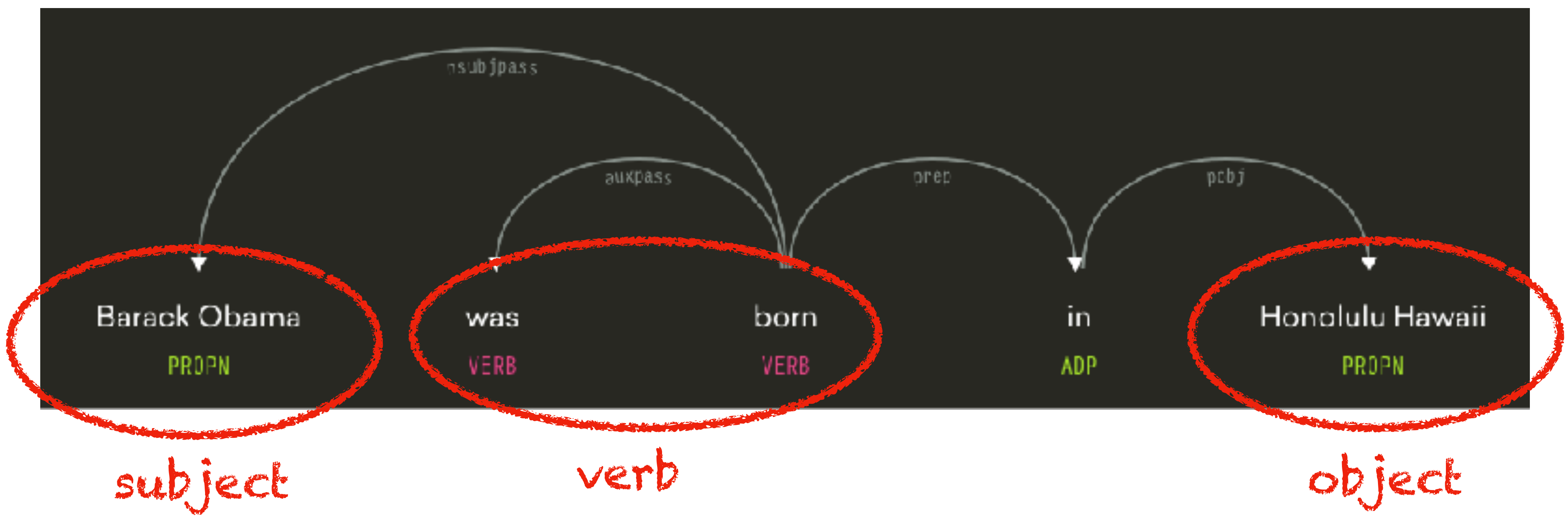
Refresher: Subject-Verb-Object Triples

Barack Obama was born in Honolulu, Hawaii.



Refresher: Subject-Verb-Object Triples

Barack Obama was born in Honolulu, Hawaii.



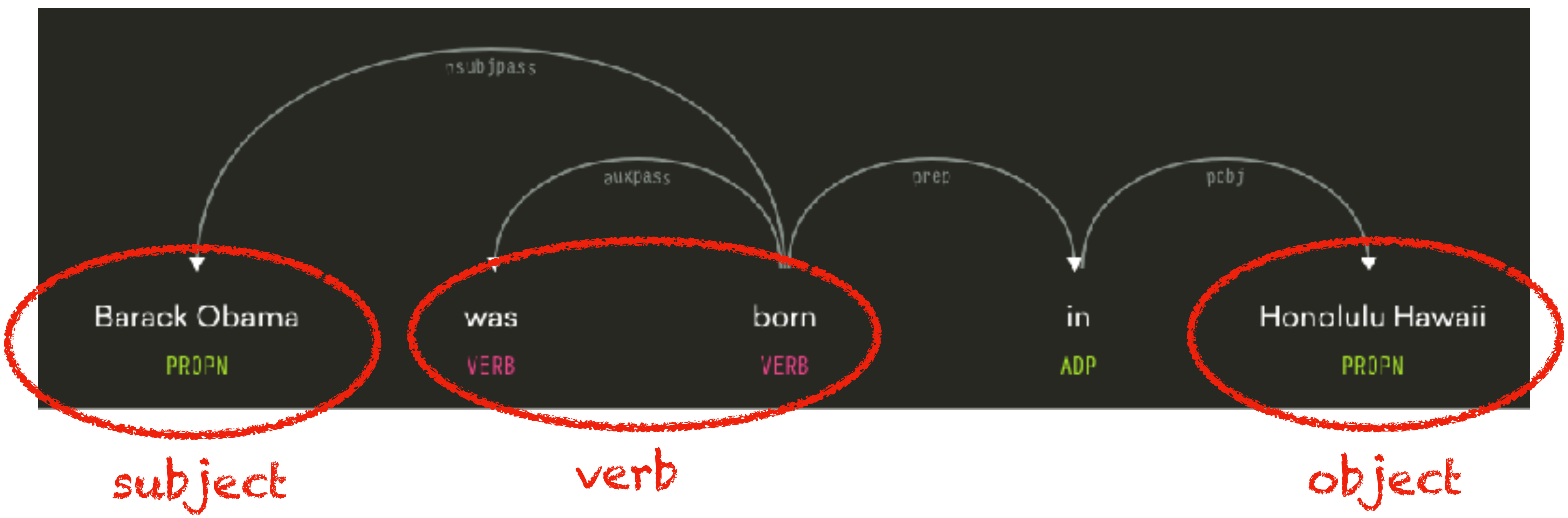
"Barack Obama" → `BarackObama`

"was born" → `PlaceOfBirth(PERSON, LOCATION)`

"Honolulu, Hawaii" → `Honolulu`

Refresher: Subject-Verb-Object Triples

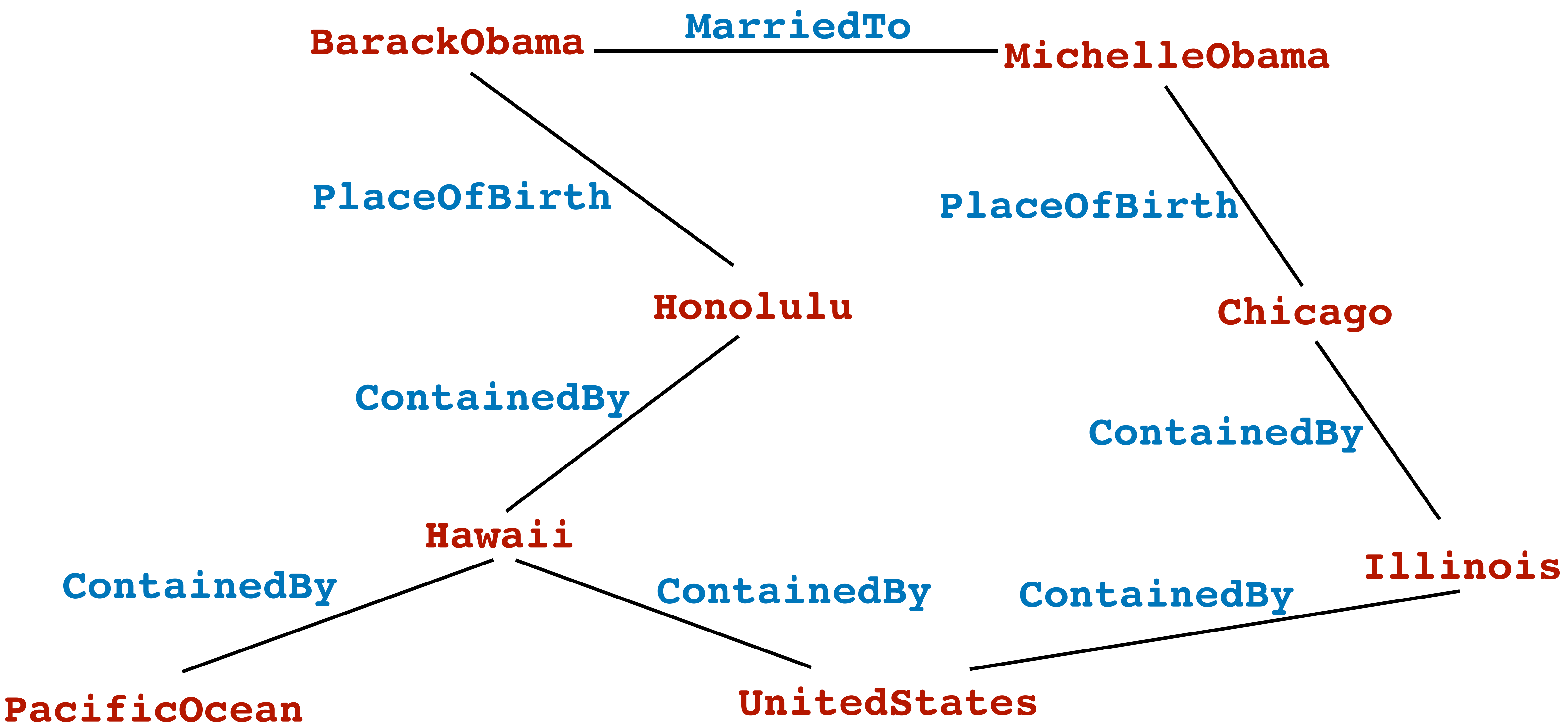
Barack Obama was born in Honolulu, Hawaii.



“Barack Obama”	“was born”	“Honolulu, Hawaii”
`BarackObama`	`PlaceOfBirth(PERSON, LOCATION)`	`Honolulu`

`PlaceOfBirth(BarackObama, Honolulu)`

Refresher: Subject-Verb-Object Triples



<u>Semantic parsing</u> maps from a natural language question to a logical form

Semantic parsing task:
Given a natural language question, map it to a logical form query that can be executed over a given database

`Q: When was Ada Lovelace born?` → `BirthYear.AdaLovelace`

`Q: What college did Obama go to?` → `Type.University ⊓ Education.BarackObama`

<u>Semantic parsing</u> maps from a natural language question to a logical form

Semantic parsing task:
Given a natural language question, map it to a logical form query that can be executed over a given database

`Q: When was Ada Lovelace born?` → `BirthYear.AdaLovelace`

`Q: What college did Obama go to?` → `Type.University ⊓ Education.BarackObama`

This example: *Lambda Dependency-Based Compositional Semantics (λ-DCS)*
(Could be any query language, e.g. SQL, SPARQL, etc.)

<u>Semantic parsing</u> maps from a natural language question to a logical form

`Q: Where was Obama born?` → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this?

<u>Semantic parsing</u> maps from a natural language question to a logical form

Q: Where was Obama born? → Type.Location ⊓ PeopleBornHere.BarackObama

How can we do this? (one example — Berant et al. 2013)

- Map natural language utterances to logical predicates

Obama → BarackObama → entity

born → PeopleBornHere → relation

where → Type.Location

PeopleBornHere(LOCATION, PERSON)

<u>Semantic parsing</u> maps from a natural language question to a logical form

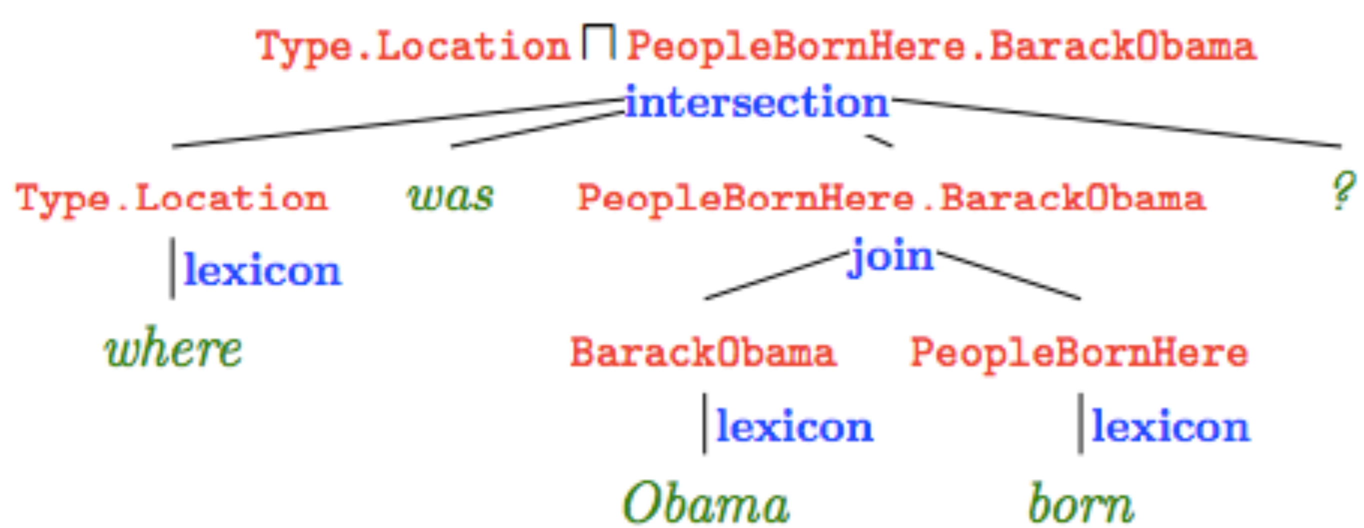
`Q: Where was Obama born?` → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Map natural language utterances to logical predicates

`Obama`	→	`BarackObama`
`born`	→	`PeopleBornHere`
`where`	→	`Type.Location`

- Compose predicates into a coherent logical form



<u>Semantic parsing</u> maps from a natural language question to a logical form

Q: Where was Obama born? → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Map natural language utterances to logical predicates

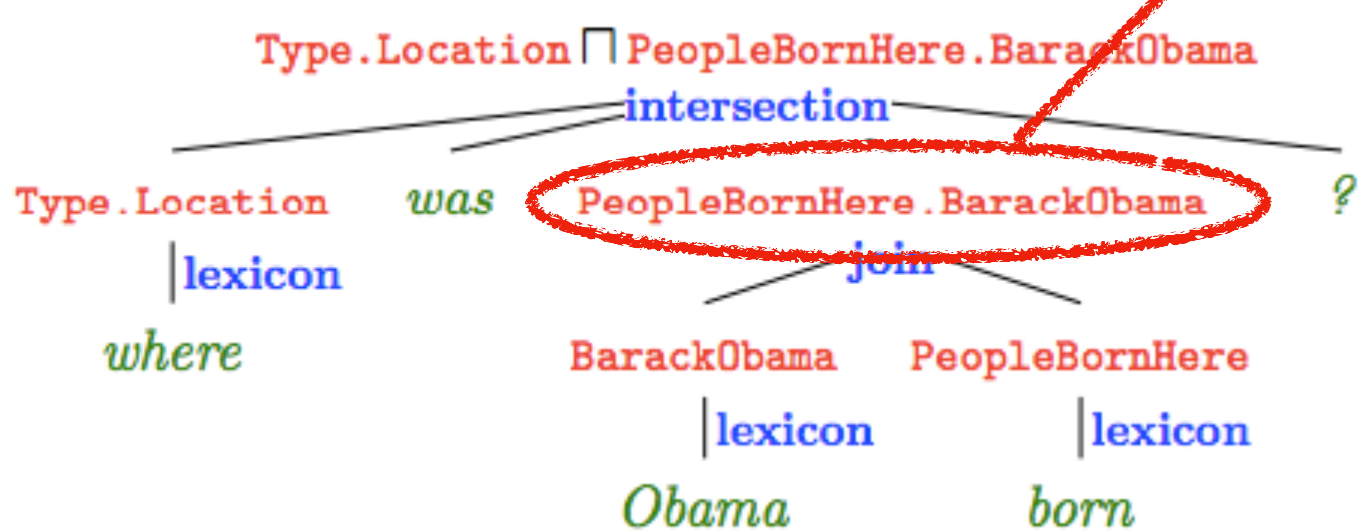
Obama → Ba

born → Pe

where → Ty

Query returns all entities *e* such that `PeopleBornHere(e, BarackObama)` exists

- Compose predicates into a coherent logical form



<u>Semantic parsing</u> maps from a natural language question to a logical form

Q: Where was Obama born? → Type.Location ⊓ PeopleBornHere.BarackObama

How can we do this? (one example — Berant et al. 2013)

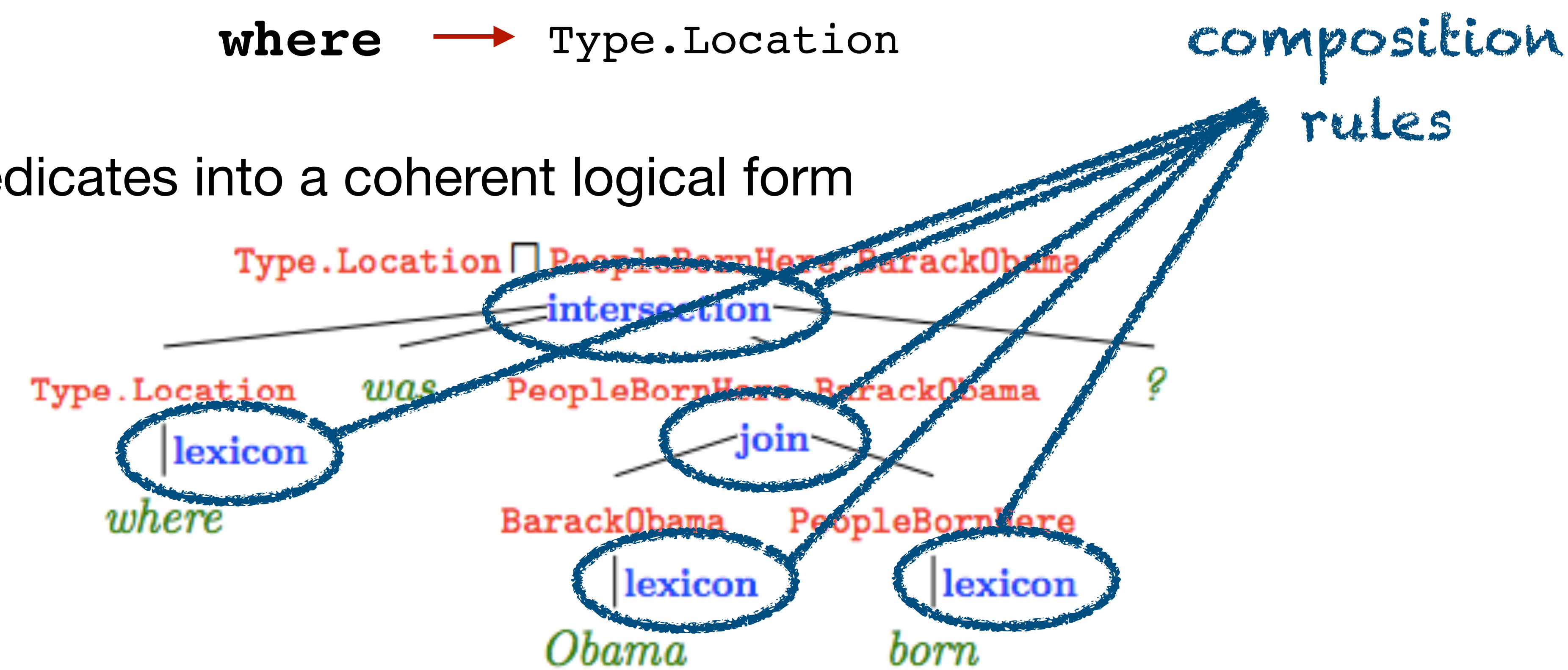
- Map natural language utterances to logical predicates

Obama → BarackObama

born → PeopleBornHere

where → Type.Location

- Compose predicates into a coherent logical form



<u>Semantic parsing</u> maps from a natural language question to a logical form

Q: Where was Obama born? → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)

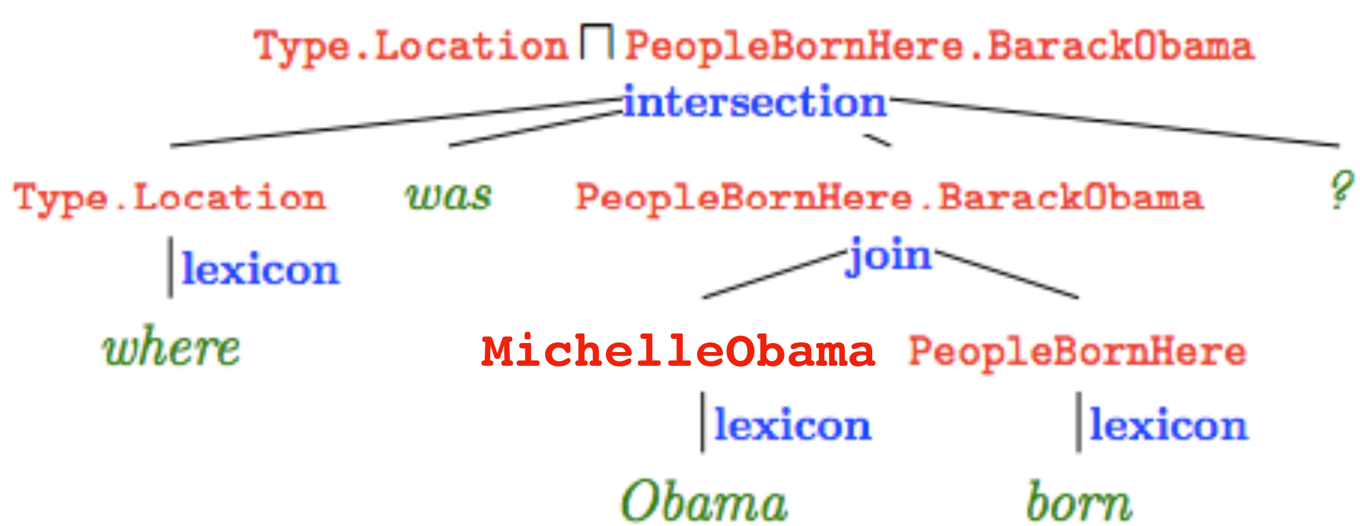
<u>Semantic parsing</u> maps from a natural language question to a logical form

`Q: Where was Obama born?` → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)
  - For each question x_i , construct set of all possible logical forms $D(x)$

ex) $d_i \in D(x)$



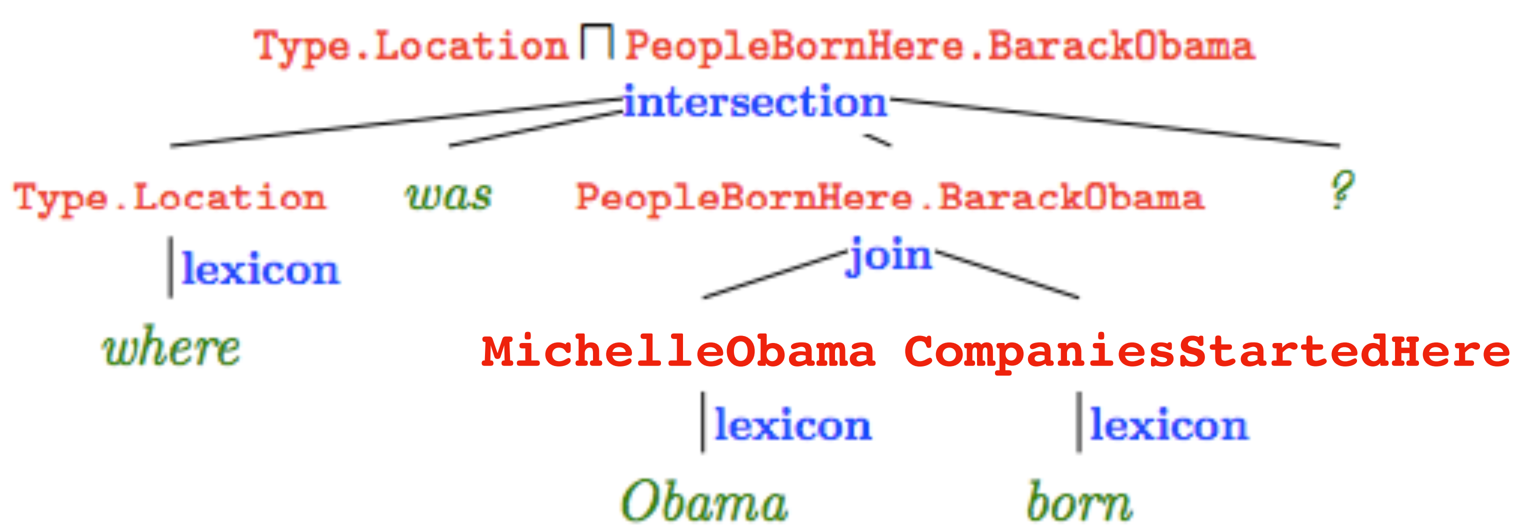
<u>Semantic parsing</u> maps from a natural language question to a logical form

Q: Where was Obama born? → Type.Location ⊓ PeopleBornHere.BarackObama

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)
  - For each question x_i, construct set of all possible logical forms $D(x)$

ex) $d_j \in D(x)$



<u>Semantic parsing</u> maps from a natural language question to a logical form

`Q: Where was Obama born?` → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)
  - For each question x_i, construct set of all possible logical forms $D(x)$
  - Model the likelihood of a derivation d given a text question x

$$p_\theta(d|x) = \frac{exp\{\phi(x,d)^\intercal\theta\}}{\sum_{d' \in D(x)} exp\{\phi(x,d')^\intercal\theta\}}$$

Semantic parsing maps from a natural language question to a logical form

`Q: Where was Obama born?` → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)
  - For each question x_i, construct set of all possible logical forms $D(x)$
  - Model the likelihood of a derivation d given a text question x

$$p_\theta(d|x) = \frac{exp\{\phi(x,d)^\intercal\theta\}}{\sum_{d' \in D(x)} exp\{\phi(x,d')^\intercal\theta\}}$$

likelihood of the derivation (tree) given the text

<u>Semantic parsing</u> maps from a natural language question to a logical form

Q: Where was Obama born? → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)
  - For each question x_i, construct set of all possible logical forms $D(x)$
  - Model the likelihood of a derivation d given a text question x

$$p_\theta(d|x) = \frac{exp\{\phi(x,d)^\top\theta\}}{\sum_{d' \in D(x)} exp\{\phi(x,d')^\top\theta\}}$$

log linear model

<u>Semantic parsing</u> maps from a natural language question to a logical form

`Q: Where was Obama born?` $\longrightarrow$ `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)
  - For each question x_i, construct set of all possible logical forms $D(x)$
  - Model the likelihood of a derivation d given a text question x

$$p_\theta(d|x) = \frac{exp\{\phi(x,d)^\intercal\theta\}}{\sum_{d' \in D(x)} exp\{\phi(x,d')^\intercal\theta\}}$$

feature function

feature weights

<u>Semantic parsing</u> maps from a natural language question to a logical form

`Q: Where was Obama born?` → `Type.Location ⊓ PeopleBornHere.BarackObama`

How can we do this? (one example — Berant et al. 2013)

- Train a model on a set of question-answer pairs (x_i, y_i)
  - For each question x_i , construct set of all possible logical forms $D(x)$
  - Model the likelihood of a derivation d given a text question x
  - Train the model such that we maximize the likelihood of derivations that give the correct answer, y . Maximize the objective:

training instances

$$\mathcal{O}(\theta) = \sum_{i=1}^{n} \log \sum_{d \in D(x) : [\![d.z]\!]_{\mathcal{K}} = y_i} p_\theta(d \mid x_i)$$

maximize likelihood of derivations that resolve to the correct answer in the knowledge base

How do we map natural language text to predicates?

“become capital of”

“political capital of”

“capital city of”

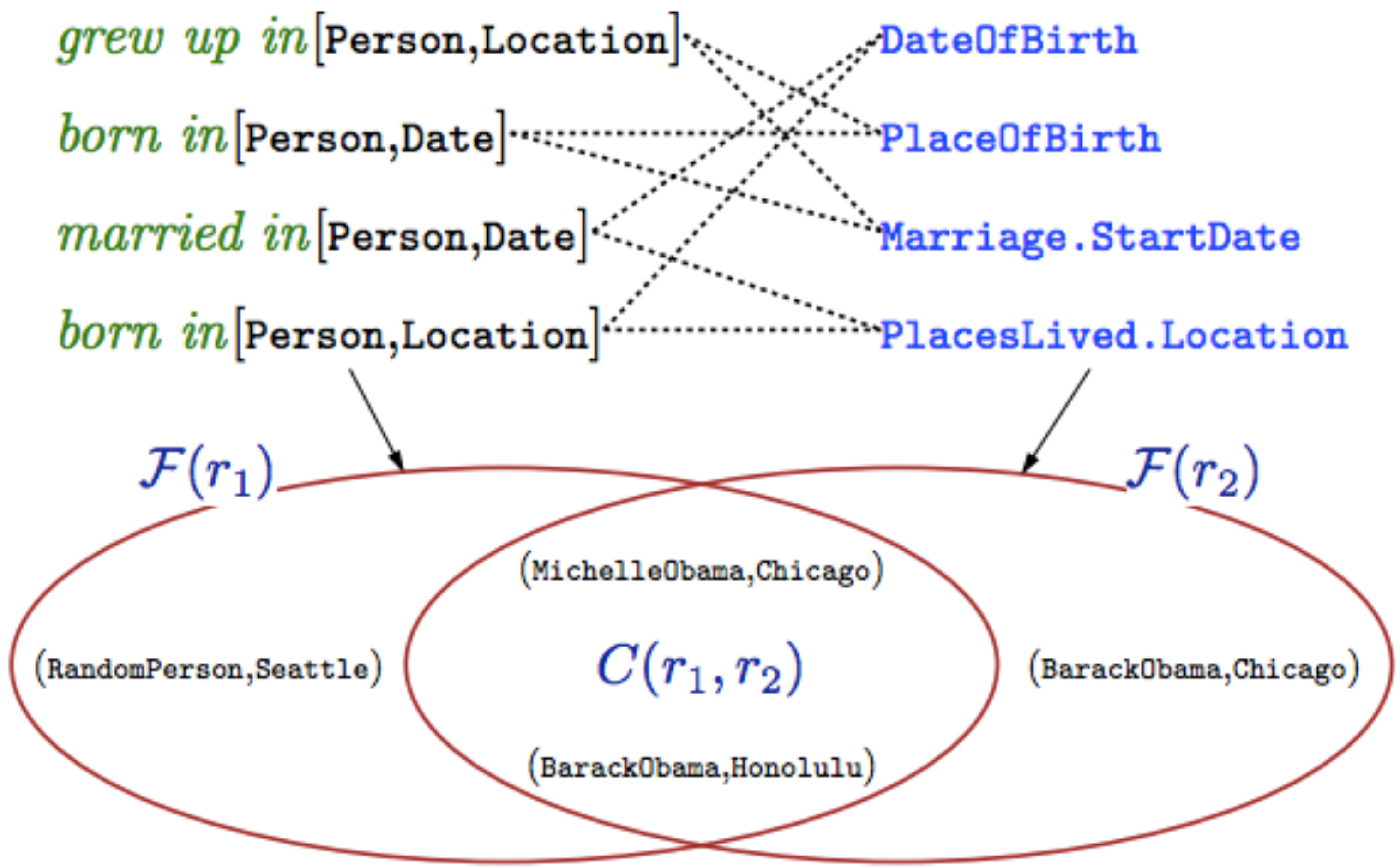
“capital city in”

“bustling capital of”

`CapitalOf(COUNTRY, CITY)`

- For limited domains, we can write rules
- Berant et al. (2013) align a large set corpus to Freebase
  - **Intuition**: a phrase and a database predicate align if they co-occur with many of the same entities

How do we map natural language text to predicates?



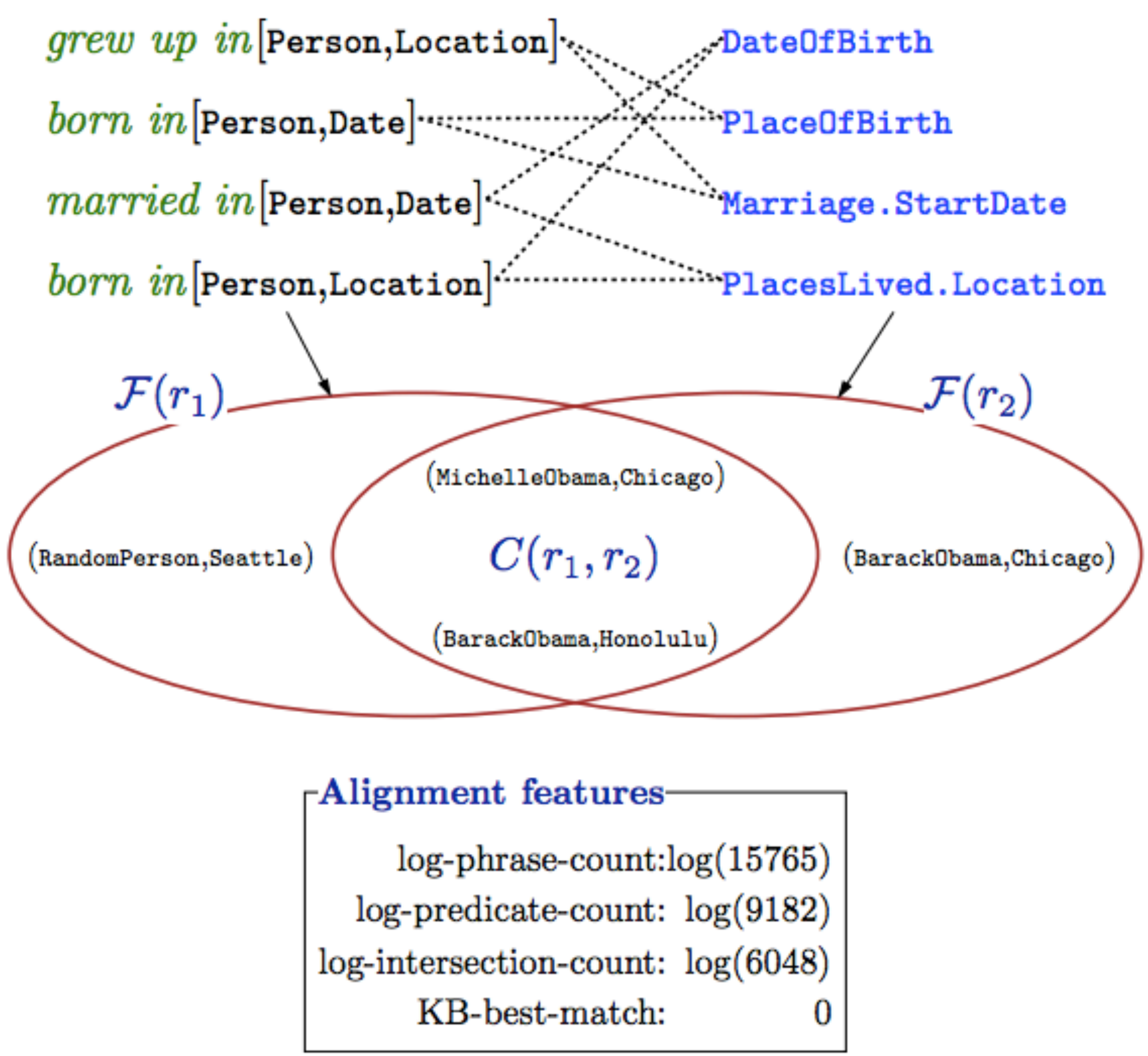
Alignment features

log-phrase-count:	log(15765)
log-predicate-count:	log(9182)
log-intersection-count:	log(6048)
KB-best-match:	0

(Berant et al. 2013)

How do we map natural language text to predicates?

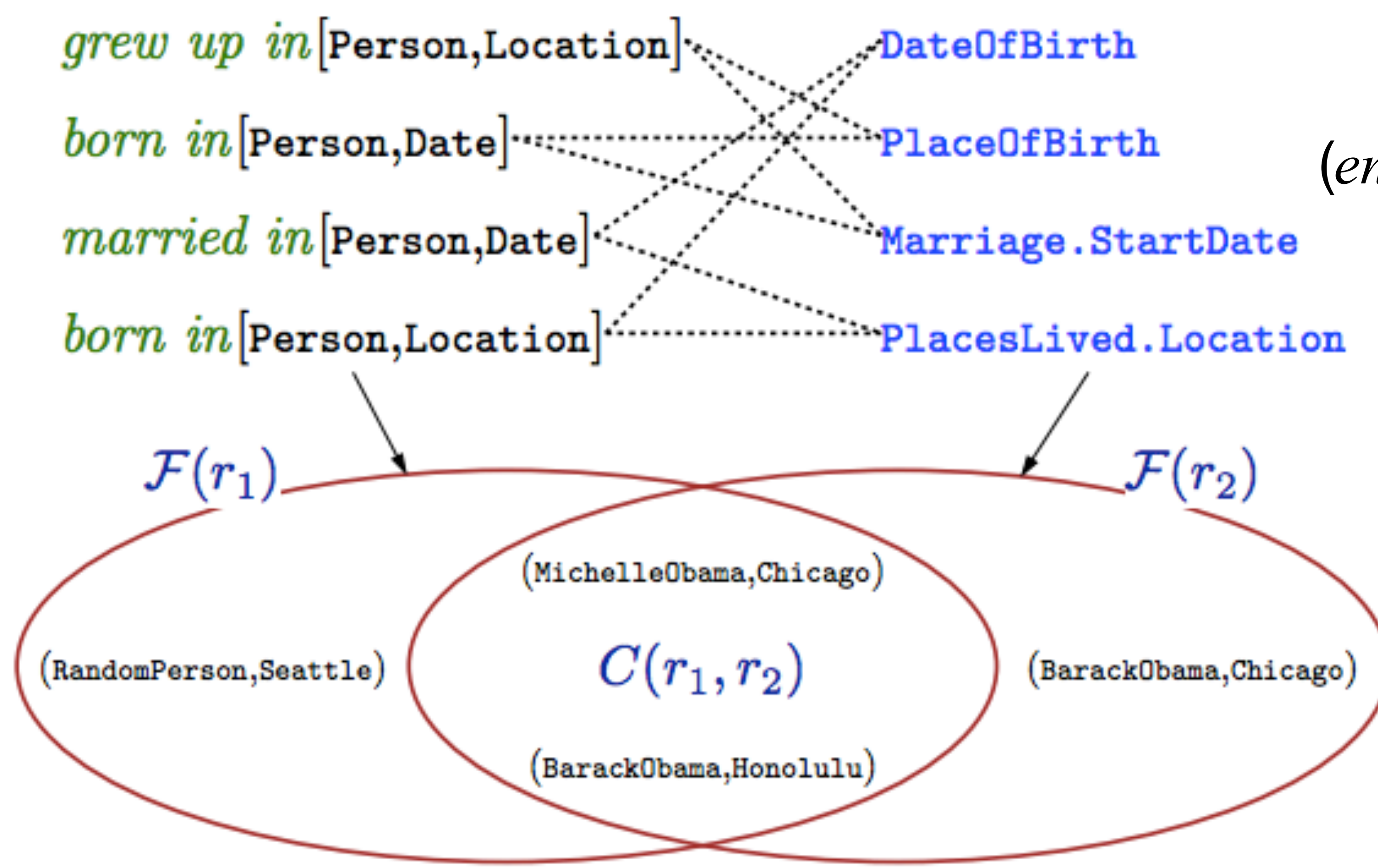
1 Extract S-V-O phrases from a corpus



(Berant et al. 2013)

How do we map natural language text to predicates?

1 Extract S-V-O phrases from a corpus



2 Extract (*entity, predicate, entity*) triples from a database

Alignment features

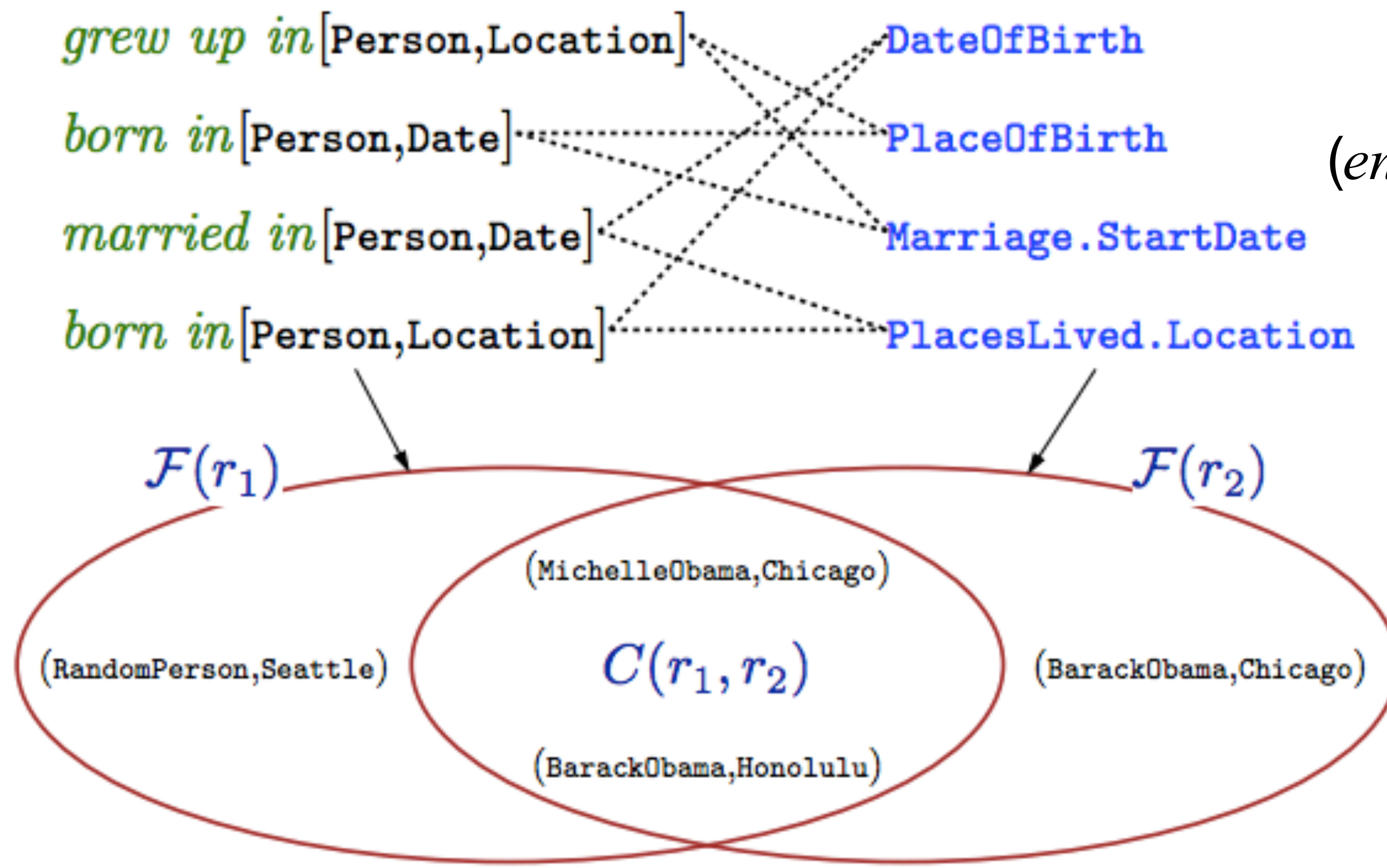
log-phrase-count:	log(15765)
log-predicate-count:	log(9182)
log-intersection-count:	log(6048)
KB-best-match:	0

(Berant et al. 2013)

How do we map natural language text to predicates?

1 Extract S-V-O phrases from a corpus

2 Extract (*entity, predicate, entity*) triples from a database



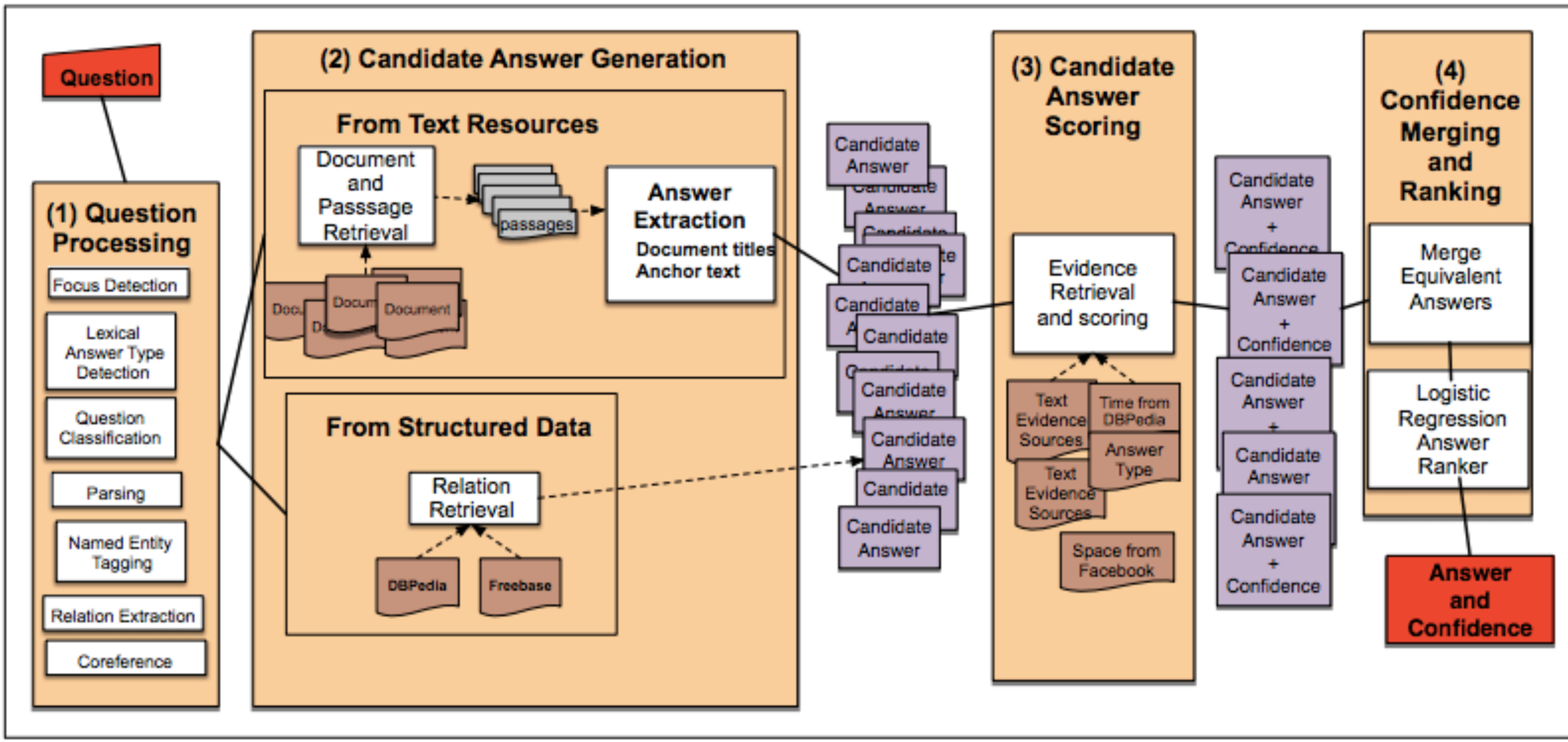
Alignment features

log-phrase-count:	log(15765)
log-predicate-count:	log(9182)
log-intersection-count:	log(6048)
KB-best-match:	0

3 Generate features based on the entities co-occurring with each

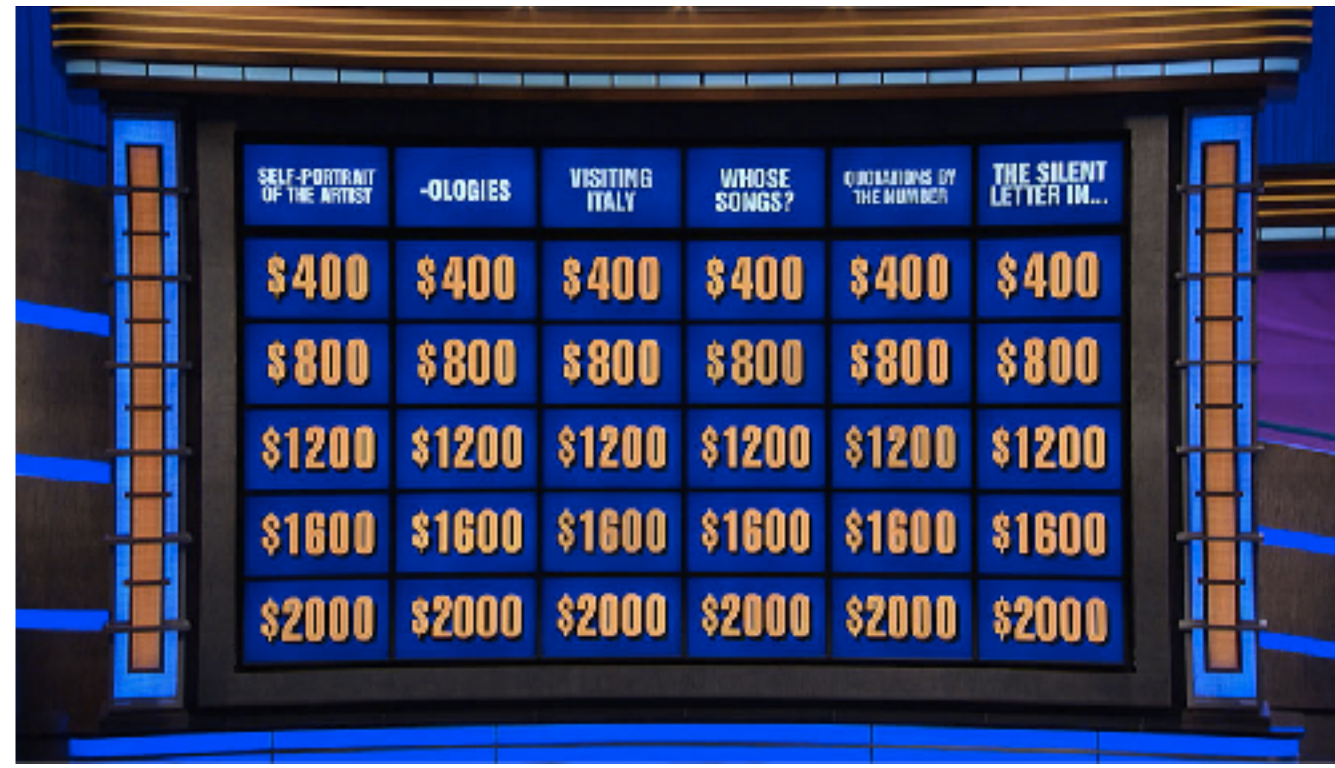
(Berant et al. 2013)

A Hybrid Approach: IBM's Watson

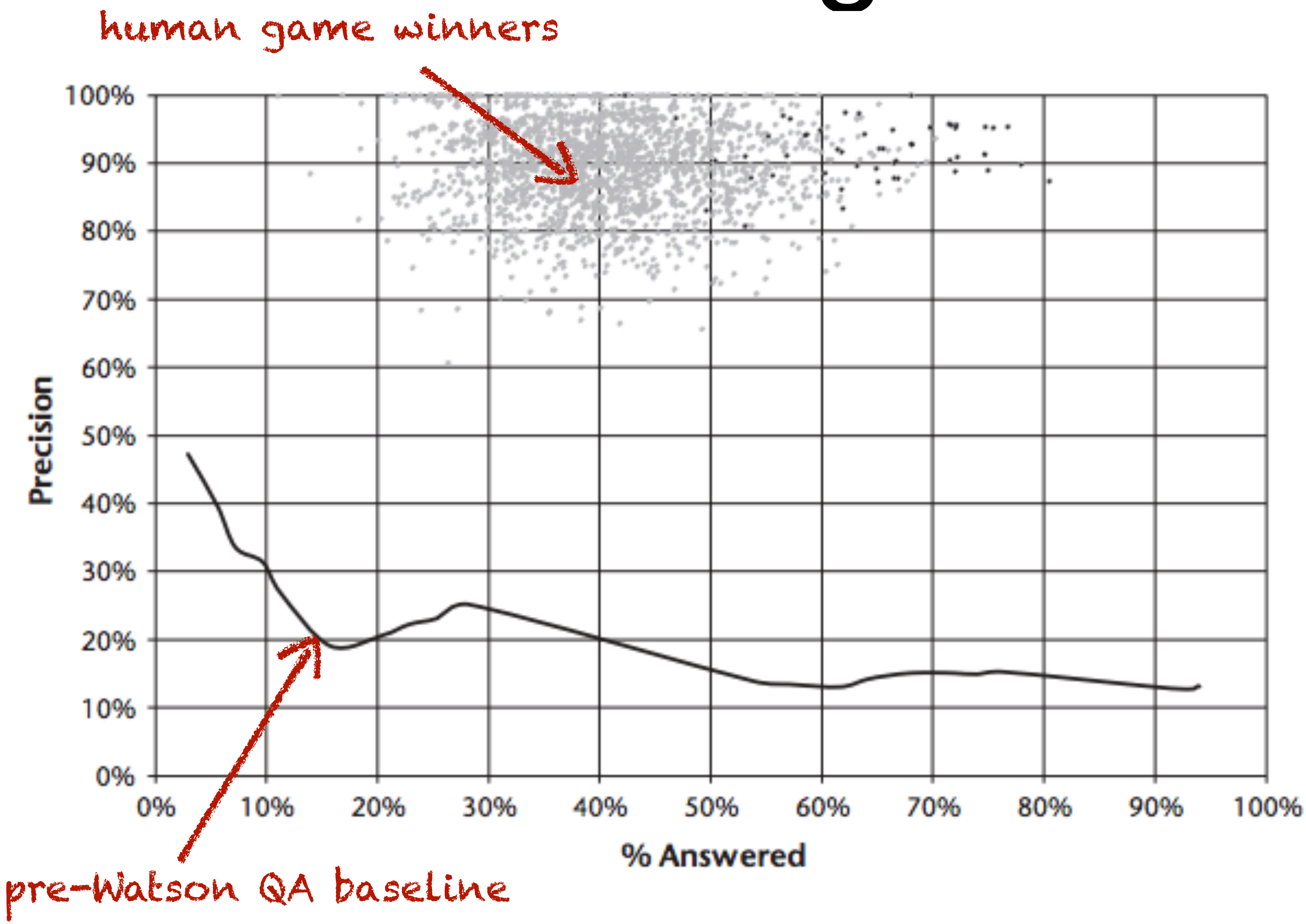


This is Jeopardy!

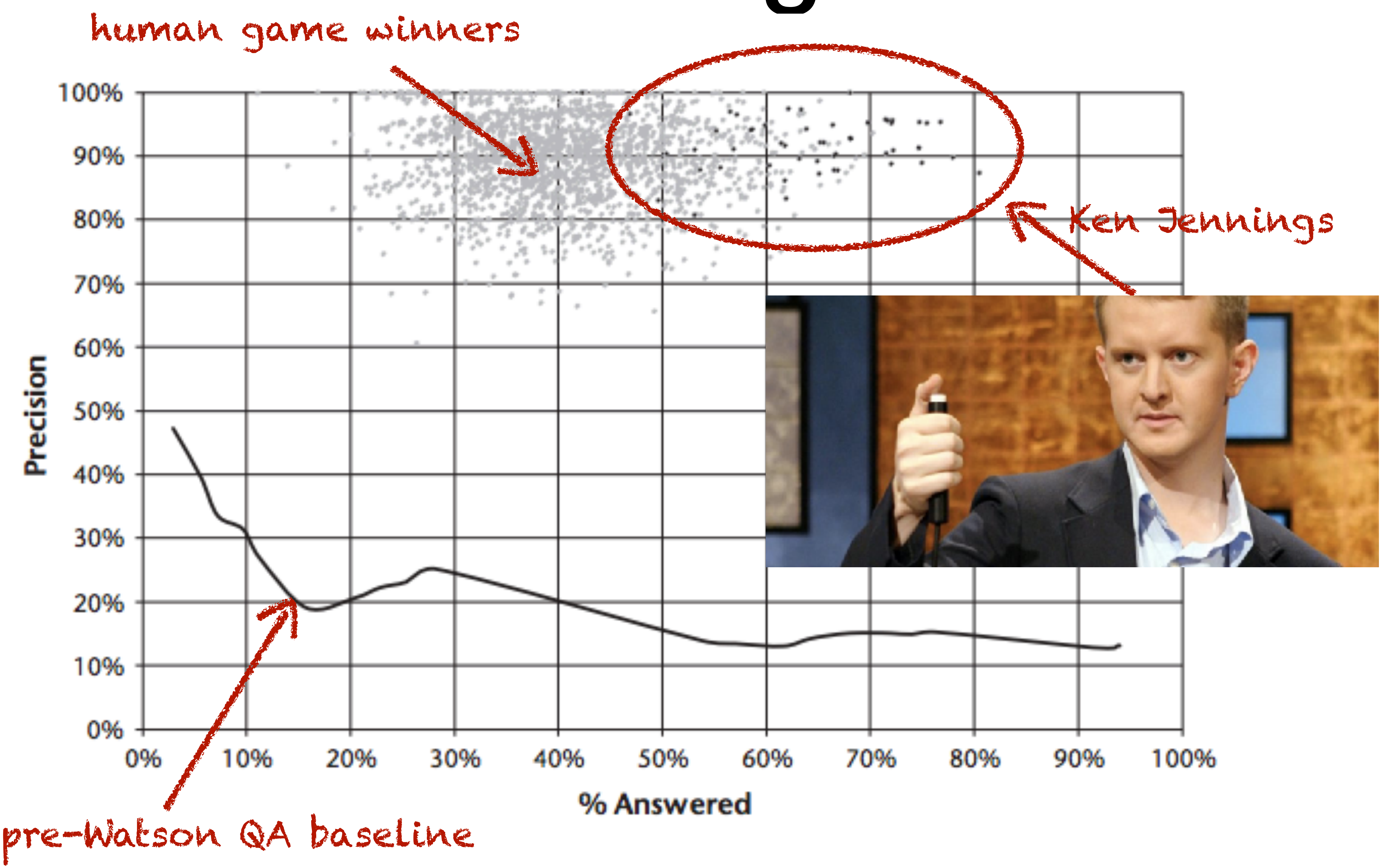
- 3 contestants
    - All must past 50-question qualifying test to be eligible
- 2 rounds of Q/A
    - One contestant chooses a category and dollar amount
    - All contestants can buzz in with answer
    - Win the dollar amount if correct; lose dollar amount for the wrong answer
- Final Jeopardy!
    - Final question, bet any amount of $$



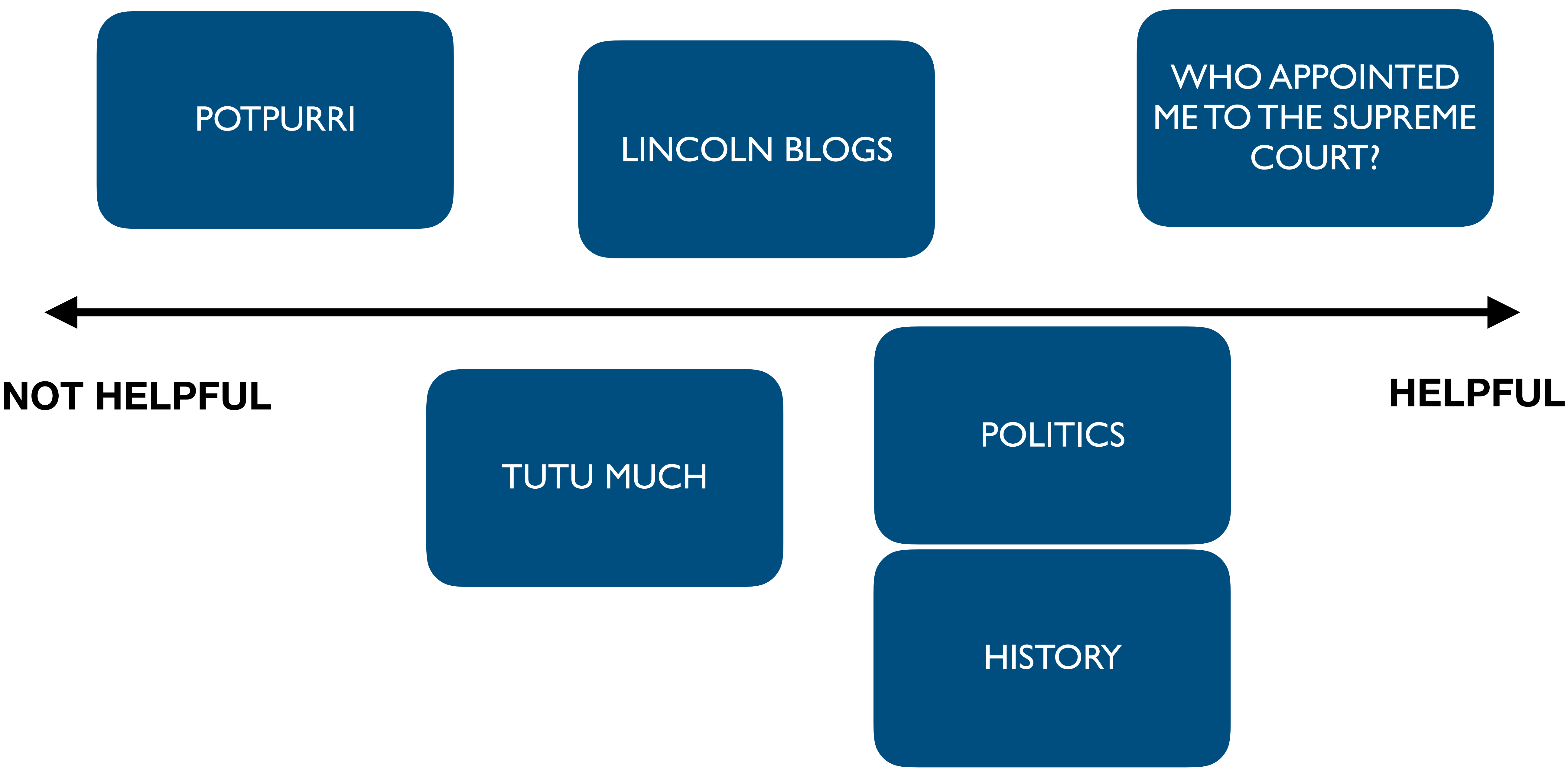
Humans are tough to beat.



Humans are tough to beat.



Why is Jeopardy so hard?
Categories (sometimes) helpful



Why is Jeopardy so hard? Different methods necessary to solve…

- Some are straightforward factoid questions

GENERAL SCIENCE

WHEN HIT BY ELECTRONS, A PHOSPHOR GIVES OFF ELECTROMAGNETIC ENERGY IN THIS FORM.

Why is Jeopardy so hard?
Different methods necessary to solve…

- Some are straightforward factoid questions

GENERAL SCIENCE

WHAT IS LIGHT?

Why is Jeopardy so hard?
Different methods necessary to solve…

- Others require DECOMPOSITION

DIPLOMATIC RELATIONS

OF THE FOUR COUNTRIES IN THE WORLD THAT THE U.S. DOES NOT HAVE DIPLOMATIC RELATIONS WITH, THE ONE THAT'S FARTHEST NORTH.

Why is Jeopardy so hard?
Different methods necessary to solve…

- Others require DECOMPOSITION

DIPLOMATIC RELATIONS

WHAT IS NORTH KOREA?

Why is Jeopardy so hard?
Different methods necessary to solve…

- Others require DECOMPOSITION

DIPLOMATIC RELATIONS

WHAT IS NORTH KOREA?

Inner subclue:

- The four countries in the world that the U.S. does not have diplomatic relations with (Bhutan, Syria, Iran, North Korea)

Outer subclue:

- Of Bhutan, Syria, Iran, North Korea, the one that's farthest north

Why is Jeopardy so hard?
Different methods necessary to solve…

- Still others are PUZZLES

RHYME TIME

IT'S WHERE PELÉ STORES HIS BALL

Why is Jeopardy so hard?
Different methods necessary to solve…

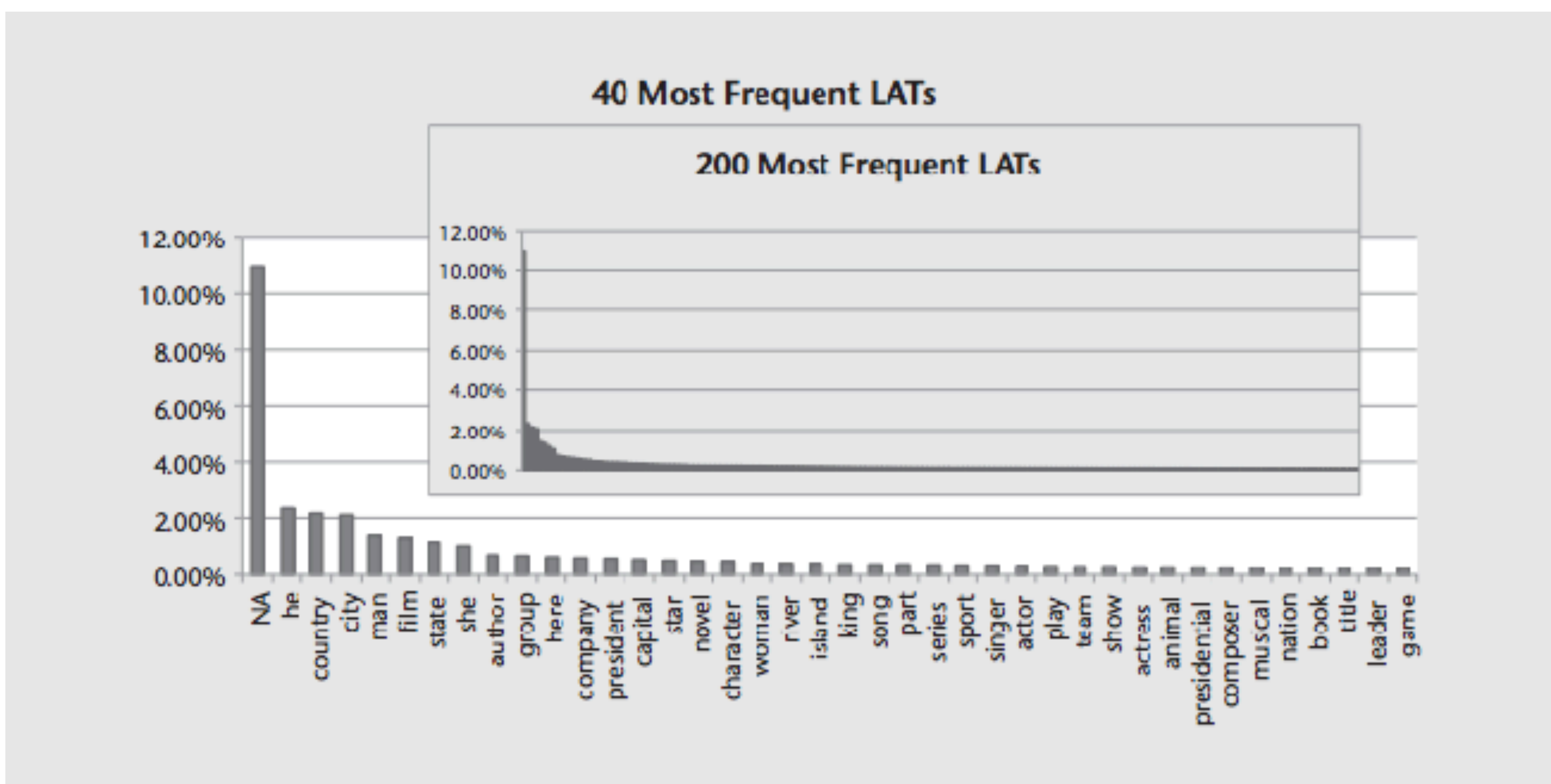
- Still others are PUZZLES

RHYME TIME

WHAT IS SOCCER LOCKER?

Why is Jeopardy so hard? The broad domain

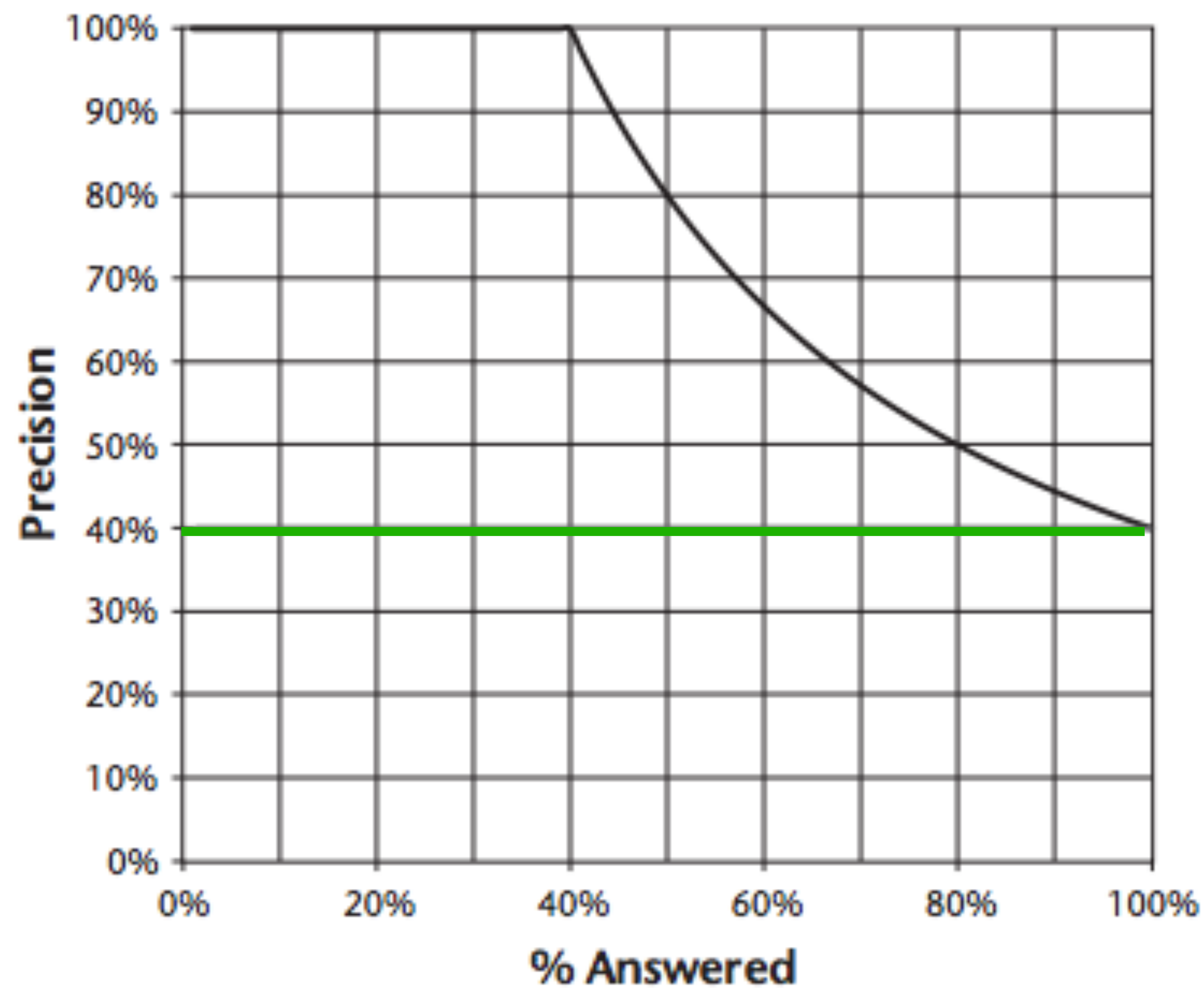
- Lexical Answer Types (LATs):
  - Category: Oooh…Chess
  - Clue: Invented in the 1500s to speed up the game, this **maneuver** involves two pieces of the same color.



Why is Jeopardy so hard? Necessity to measure confidence

- Answering a question wrong can be very costly (lose $$)
- Accurate confidence estimation is critical

Two systems with 40% accuracy:



Watson's winning approach: Solve using many systems in parallel

- Used more than 100 techniques for the various stages of QA
- **Massive parallelism:**
  - Consider multiple interpretations and hypotheses
- **Pervasive confidence estimation**
  - No component commits to an answer; all produce features and associated confidences
- **Integrate shallow and deep knowledge**
  - Balance use of strict and shallow semantics, leveraging many loosely formed ontologies

Watson's winning approach: Question Analysis

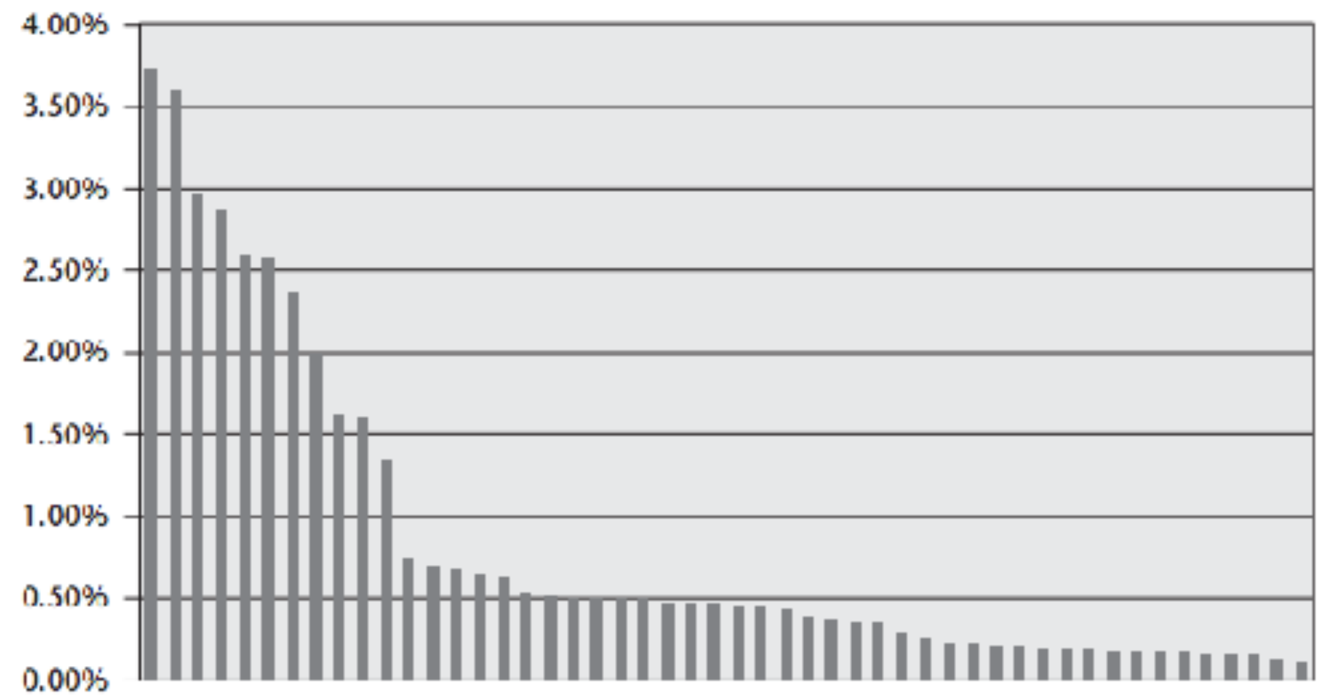
- Focus/Lexical Answer Type (LAT) detection guides IR-based QA systems
  - ex) When hit by electrons, a phosphor gives off electromagnetic energy in **this form.**
  - Focus contains useful information; is often the subject or object of a relation in the clue; can turn the question into a factual statement when replaced with a candidate

Watson's winning approach: Question Analysis

- Limited ability to use existing databases in a strict knowledge-based approach

  - Use of databases depends on ability to detect relations covered by the databases

  - Very flat distribution of Freebase relations in sample of 20k Jeopardy questions



Distribution of 50 most-frequent Freebase relations in Jeopardy questions

Watson’s winning approach: Hypothesis generation

- Initially generate several hundred candidate answers; focus primarily on **recall**
- Run a ‘soft filter’ to prune list down to ~100
- Retrieve further evidence about top-100 candidate answers
- Score each candidate using a variety of features

Watson's winning approach: Scoring candidates

HE WAS PRESIDENTIALLY PARDONED ON SEPTEMBER 8, 1974

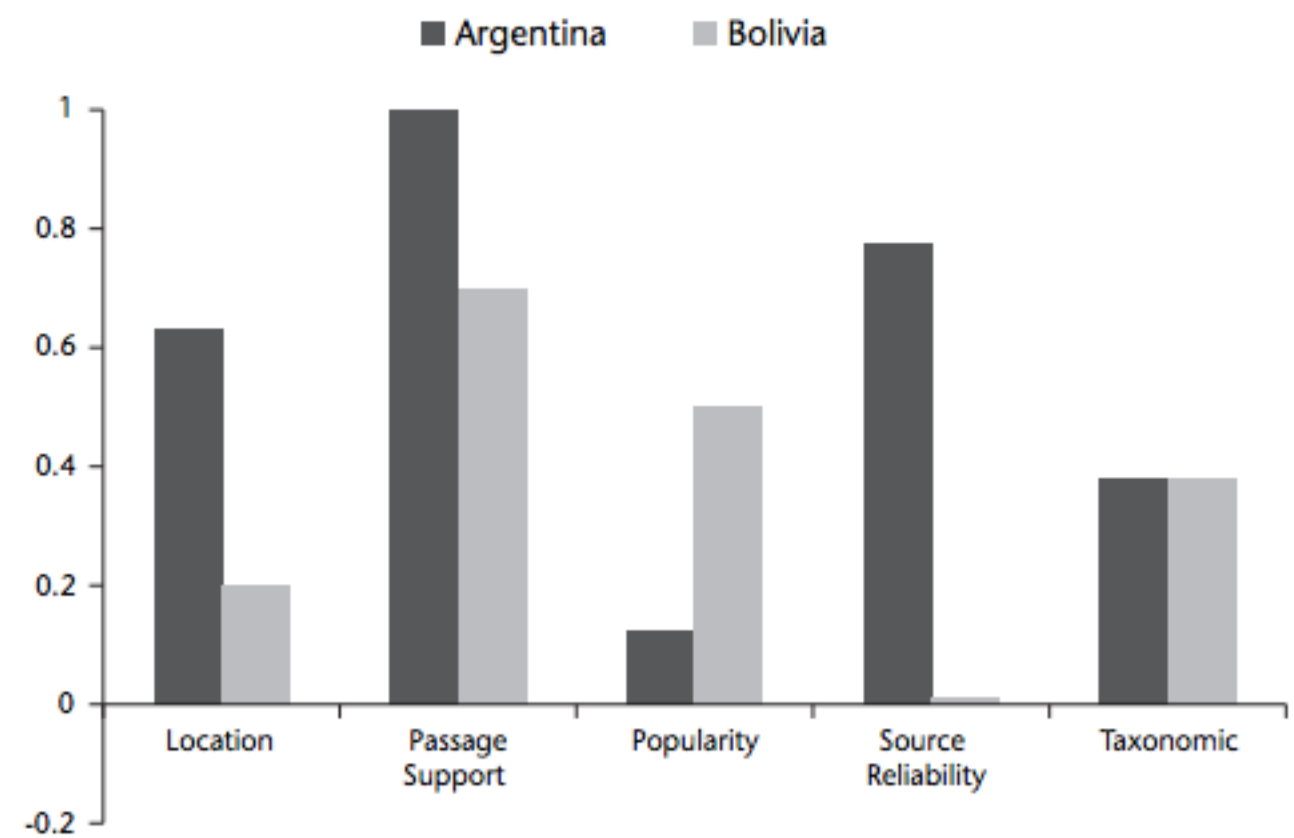
Correct answer: Nixon

Retrieved passage:
Ford pardoned Nixon on Sep. 8, 1974.

- Examples of scoring features:
  - Terms in common between clue and passage
  - Sequence matching
  - Logical form alignment
    - Aligns dependencies in clue and retrieved passage
  - (Geospatial and temporal reasoning)

Watson's winning approach: Scoring candidates

CHILE SHARES ITS LONGEST LAND BORDER WITH THIS COUNTRY.

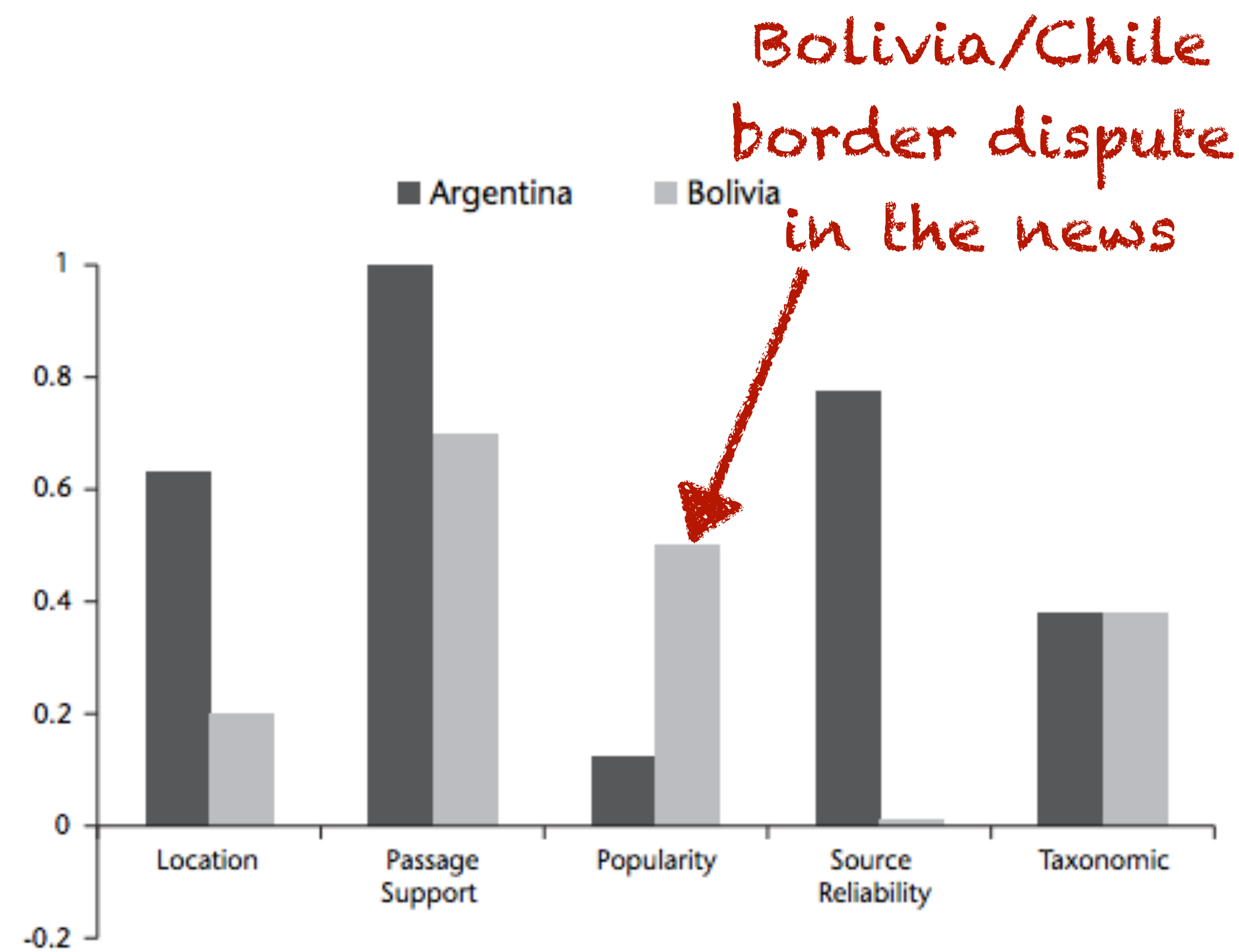


Correct answer: ARGENTINA

Evidence profiles for two candidate answers

Watson's winning approach: Scoring candidates

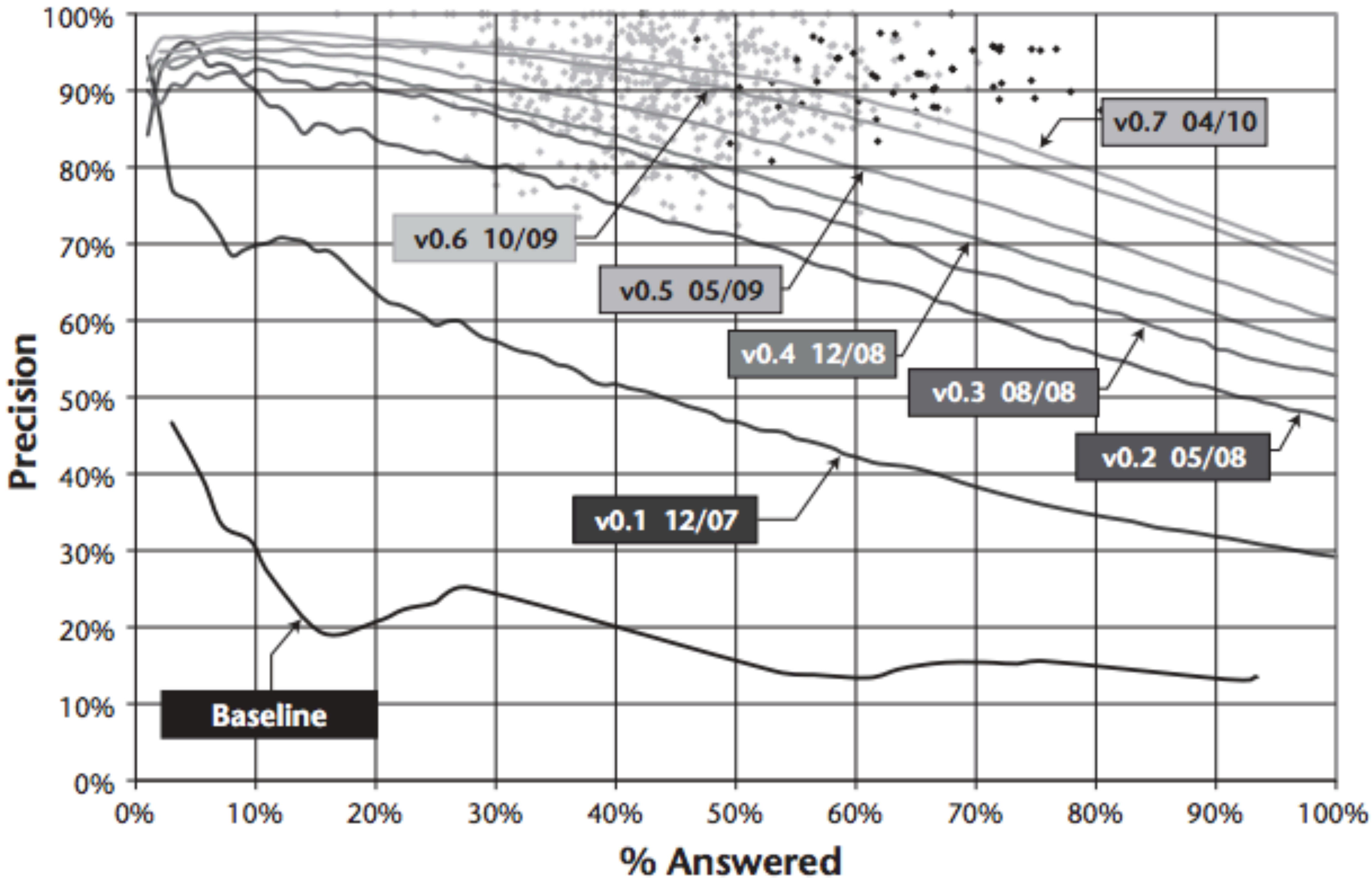
CHILE SHARES ITS LONGEST LAND BORDER WITH THIS COUNTRY.



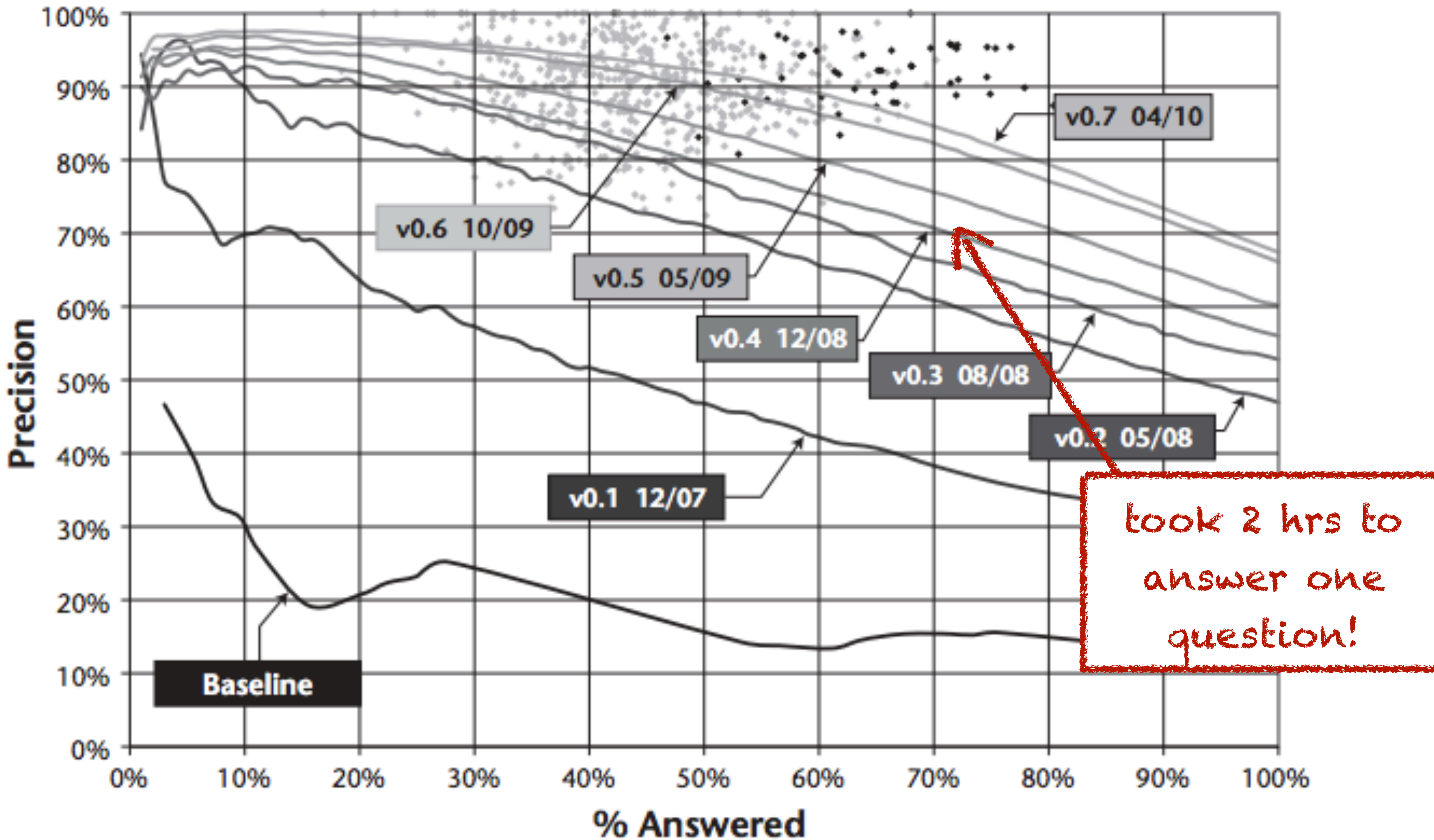
Correct answer: ARGENTINA

Evidence profiles for two candidate answers

Watson learned a lot from 2007-2010.



Watson learned a lot from 2007-2010.



Watson U.S. Cities (2:45)

Question:

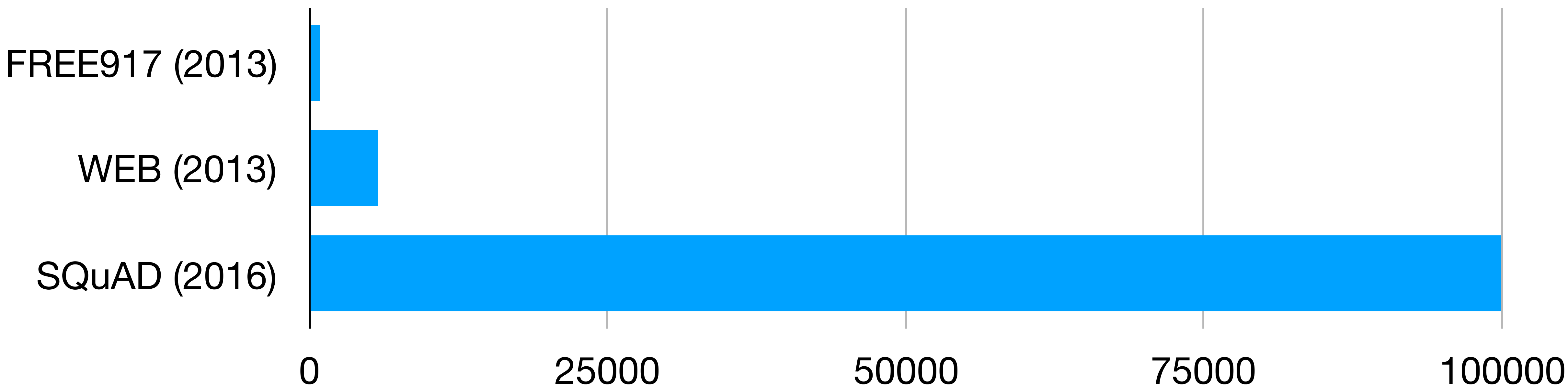
What are some things that might have gone wrong here?

(Chat with the person next to you, 2 mins)

Category	U.S. Cities
Clue	Its largest airport is named for a World War II hero; its second largest, for a World War II battle
Watson's Answer	Toronto
Correct Answer	Chicago (O'Hare and Midway airports)

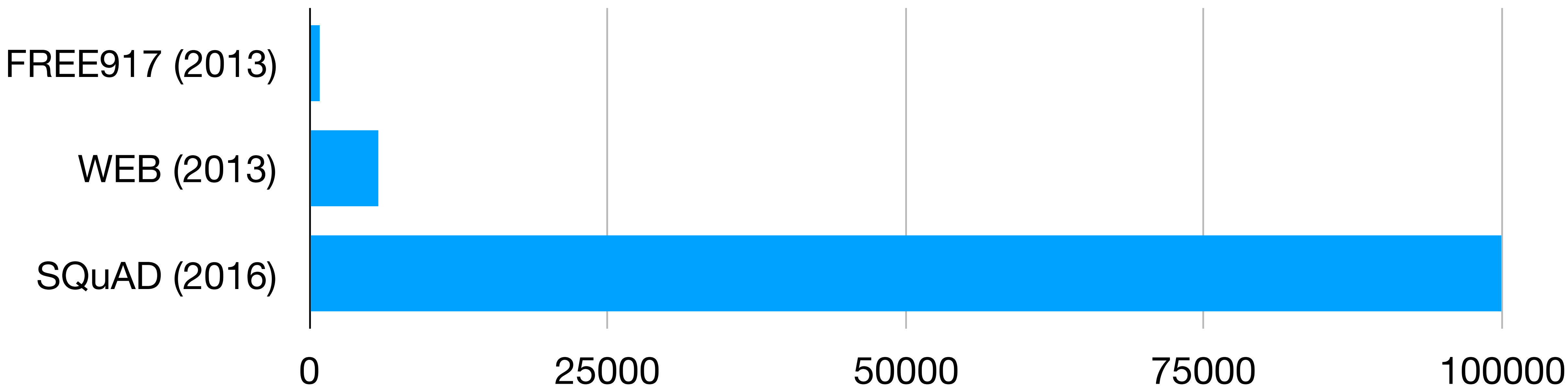
- *"It involves what we think of as 'number of hops' from the answer, so you have something missing that Jeopardy wants you to think of…"*
- *"…there are lots of cities in the United States named Toronto."*
- *"The actual city in Canada also has a baseball team in the American League…"*
- *"Very often there are categories in Jeopardy, but the answers to the questions are <u>not</u> that type of thing that's in the category. Watson knows that and says, 'Yes, U.S. city <u>might</u> be part of answering this question, but there are other clues that must be considered.'"*

The size of QA training sets has increased significantly in the past few years…



- Stanford Question Answering Dataset (SQuAD)
  - Crowdsourced 100k+ questions and answers from Wikipedia
  - Reading-comprehension-style QA

The size of QA training sets has increased significantly in the past few years…



Passage: Tesla later approached Morgan to ask for more funds to build a more powerful transmitter. **When asked where all the money had gone, Tesla responded by saying that he was affected by the Panic of 1901**, which he (Morgan) had caused. Morgan was shocked by the reminder of his part in the stock market crash and by Tesla's breach of contract by asking for more funds. Tesla wrote another plea to Morgan, but it was also fruitless. Morgan still owed Tesla money on the original agreement, and Tesla had been facing foreclosure even before construction of the tower began.

Question: On what did Tesla blame for the loss of the initial money?

Answer: Panic of 1901

Table 1: An example from the SQuAD dataset.

SQuAD Leaderboard (as of this morning)

Leaderboard

Since the release of our dataset, the community has made rapid progress! Here are the ExactMatch (EM) and F1 scores of the best models evaluated on the test set of v1.1. Will your model outperform humans on the QA task?

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Jan 22, 2018	Hybrid AoA Reader (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	82.482	**89.281**
1 Mar 06, 2018	QANet (ensemble) *Google Brain & CMU*	82.744	89.045
1 Feb 19, 2018	Reinforced Mnemonic Reader + A2D (ensemble model) *Microsoft Research Asia & NUDT*	**82.849**	88.764
2 Jan 05, 2018	SLQA+ (ensemble) *Alibaba iDST NLP*	82.440	88.607
3 Feb 02, 2018	Reinforced Mnemonic Reader (ensemble model) *NUDT and Fudan University* https://arxiv.org/abs/1705.02798	82.283	88.533
3 Feb 27, 2018	QANet+ (single model) *Google Brain & CMU*	82.209	88.608
3 Jan 03, 2018	r-net+ (ensemble) *Microsoft Research Asia*	82.650	88.493
4 Dec 22, 2017	AttentionReader+ (ensemble) *Tencent DPDAC NLP*	81.790	88.163

Top systems employ end-to-end neural network architecture

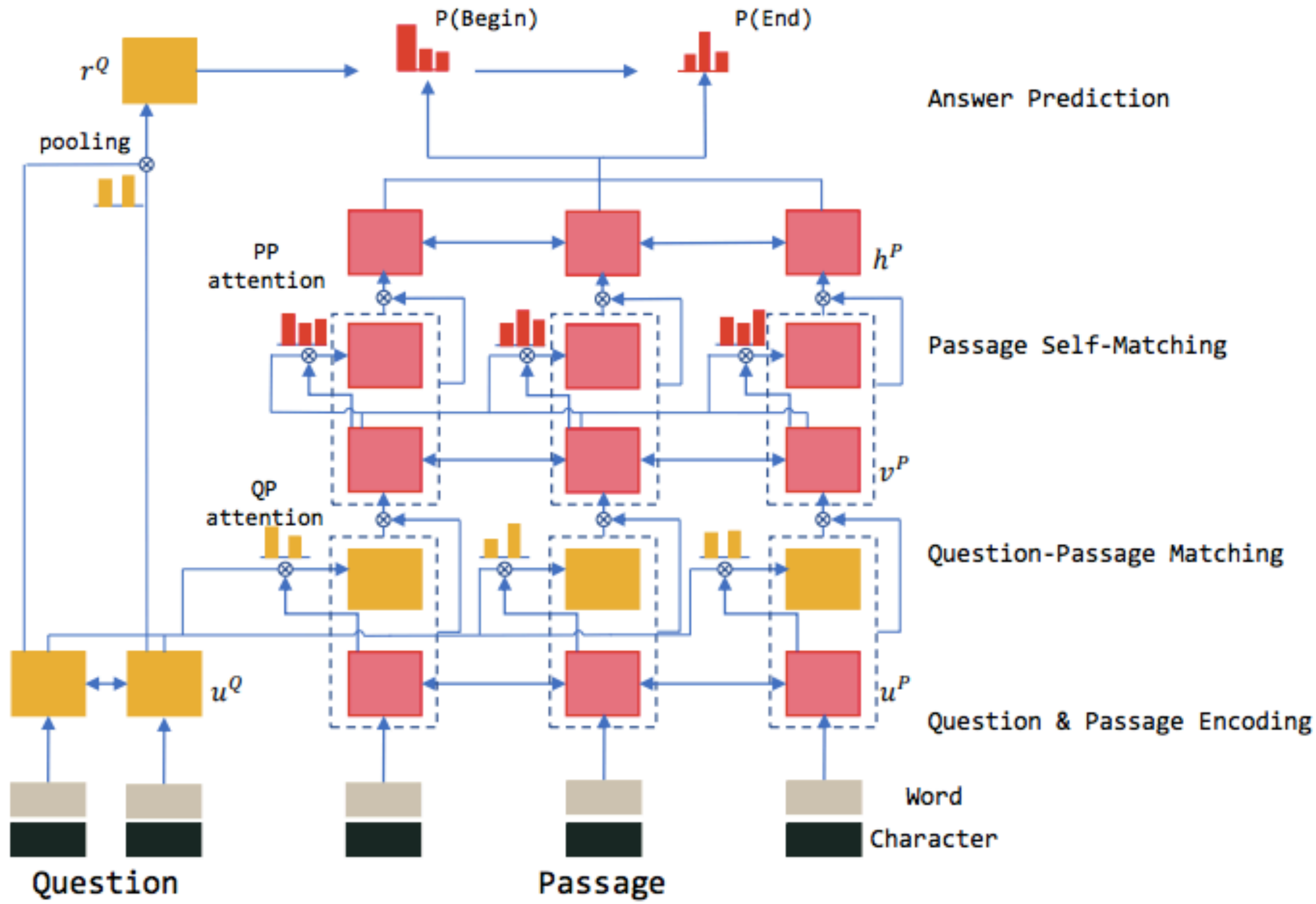


Figure 1: R-NET structure overview.

(MSR Asia, 2017)

<u>Aristo demo</u>

QUESTIONS?