Modeling questions and responses

Lecture 3: The architecture of a Question Answering (QA) system

Kyle Rawlins

NASSLLI 2016, 13th July, 2016

Johns Hopkins University, Department of Cognitive Science

Outline

More on question representations in QA

- More on unstructured topic models
- Questions as queries

A detailed look at an end-to-end system: Watson

Question analysis

Search / hypothesis generation

Confidence

Formulating an answer

Course structure

- *Lecture 1*: Introducing questions and responses.
- *Lecture 2*: Representing question meanings.

⇒ *Lecture 3-4*: The architecture of a QA system.

- *Lecture 4-5*: a pragmatics for responses.
- *Lecture 5*: wrap-up.

Decomposing Question Answering (QA)

1. Give some question Q, what information could in principle answer Q?
2. Given some possibilities for information that addresses Q, what information *correctly* answers Q?
3. Given information that correctly answers Q, how can this information be formulated as *an answer*?

Decomposing QA

Moldovan & Surdeanu (2003) fig 4:

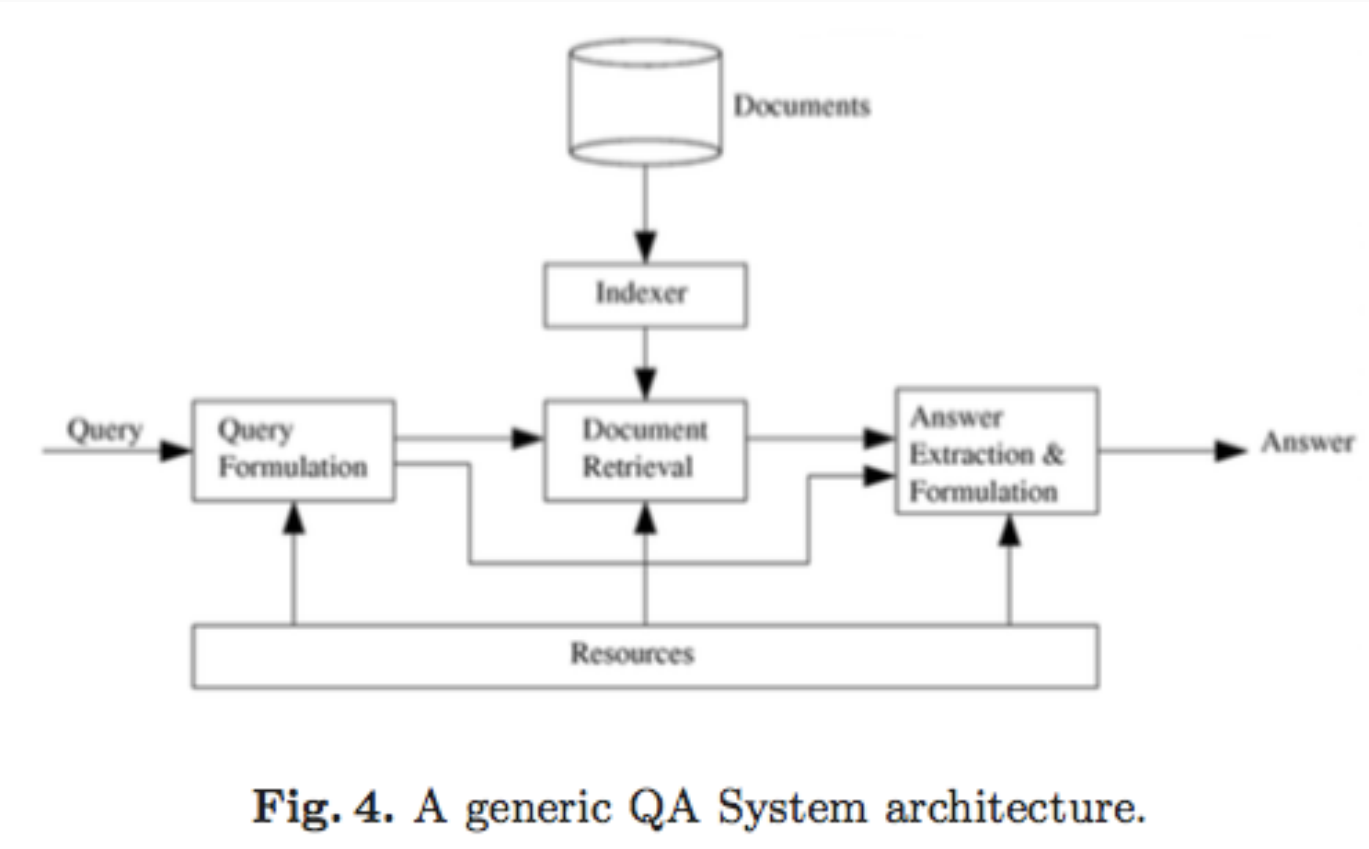


Fig. 4. A generic QA System architecture.

More on question representations in QA

Three notions of topic

What is the topic of a question? What is a question about?

- Alternative semantics: a question is about the set of issues that are related (at least, non-orthogonal; Lewis 1988) to its denotation. (Rawlins 2013, Yablo forthcoming)
- Structured meanings: a question is about the abstracted set. Or, the set of entities that could satisfy the property. Or, the property (in some sense).
- Topic models: a question is about a weighted mixture of topics constructed from content words of English.

The topic model intuition (details in previous slide set)

If you look at a large body of text, documents that share topics will tend to use the same words.

- LDA (etc) are algorithms that find these clusters of words.
- (Can run them on things other than unigram frequency, but there's an issue of sparsity.)
- Healthtrust fiction 1922-24 topic browser: `http://jgoodwin.net/htb/`

Aboutness in linguistics vs. CS

In linguistics/philosophy, notions of topic/aboutness are typically stated over semantic representations.
In CS, stated over linguistic forms.

Aboutness in linguistics vs. CS

In linguistics/philosophy, notions of topic/aboutness are typically stated over semantic representations.
In CS, stated over linguistic forms.

- Assumption: in aggregate sentences(/documents) that share content words will be 'about' the same thing.
- Linguistic forms (given enough data) are a proxy for a deeper semantics?
- Computational semantics, over time, uses more and more sophisticated versions of 'linguistic form'. Universal dependency parse: hybrid syntactic/semantic parse.

A very old idea: questions as db queries

Suppose you have a SQL table that looks like this:

(1) Table `People`:

firstname	lastname	birthdate	birthplace	height	...
Justin	Trudeau	12/25/71	Ottawa	6’ 2”	...
Kyle	Rawlins	12/30/79	Boston	5’ 10”	...
...					

A very old idea: questions as db queries

Suppose you have a SQL table that looks like this:

(1) Table `People`:

firstname	lastname	birthdate	birthplace	height	...
Justin	Trudeau	12/25/71	Ottawa	6' 2"	...
Kyle	Rawlins	12/30/79	Boston	5' 10"	...
...					

Then, if you can convert 'When was JT born?' into the following, you're all set:

```
from People select birthdate where first-
name="Justin" and lastname="Trudeau";
```

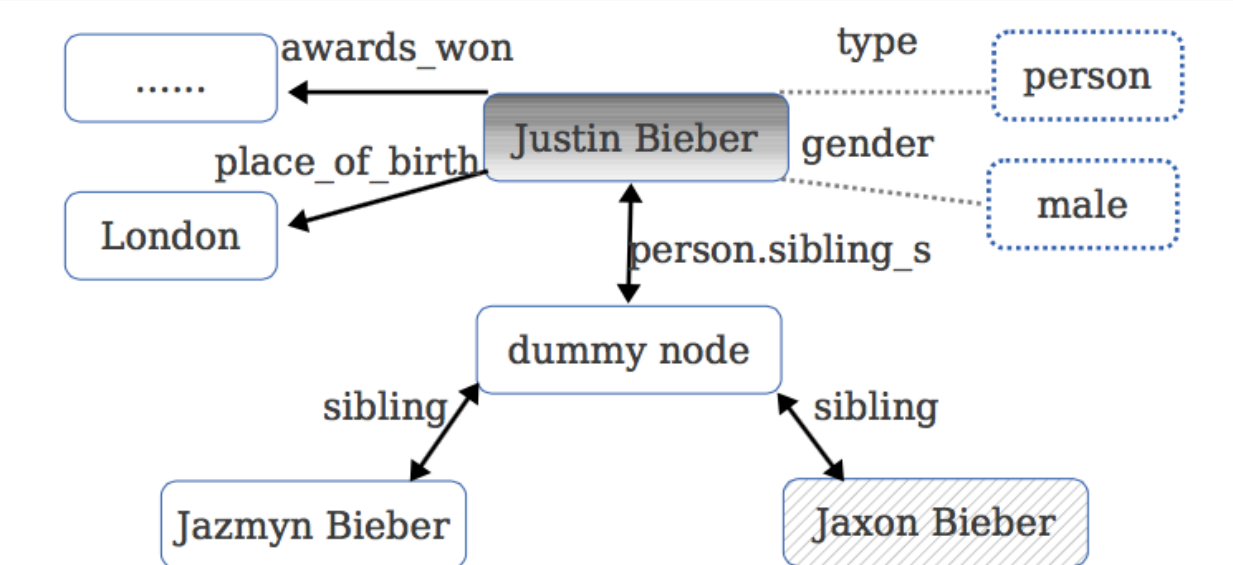
Knowledge-base queries

In fact, many organizations are collecting just this sort of structured information. (e.g. Freebase ⇒ Google Knowledge Graph)

Converting NL to queries

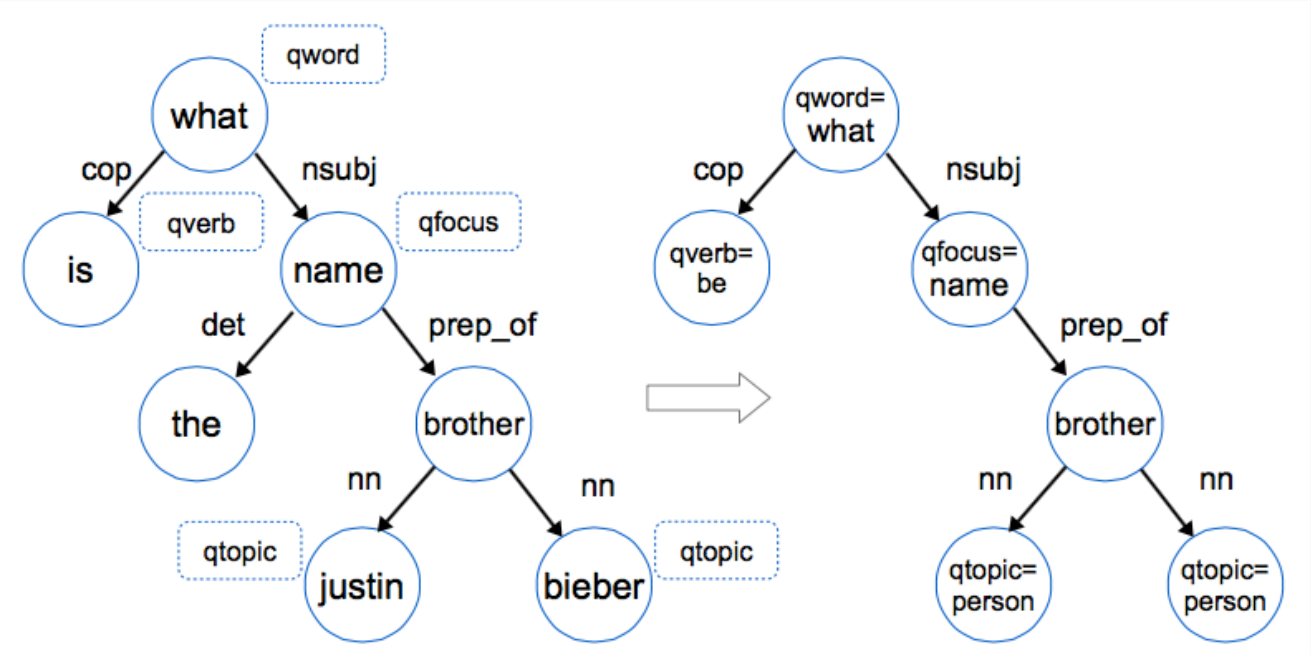
Getting the question ⇒ query mapping in general is extremely hard.

- Traditional databases are much too rigid to do this in a general way.
- Need probabilistic models, flexible data / query language.
- Modern viewpoint: query/knowledgebase as semi-structured topic graphs (Yao & Durme 2014). Freebase:



Convert NL to queries

Yao & Durme (2014) query format: convert a Stanford dependency parse to a more general graph:



Learn a model for aligning question graphs to topic graphs. (A lot of hard machine learning elided here.)

Overview of Yao's system

Core resource: preprocessed freebase (treat as a graph).

- Use ClueWeb09 (5TB of web pages) as a training set to learn associations between Freebase relations and text.

1. Query analysis: a. universal dependency parse. b. abstract away syntactic details to a topic graph.
2. Search: probabilistically match topic graph to freebase using the learned associations.
3. Answer formulation: just present the freebase node.

Summarizing

- All four types of question-meaning discussed today/yesterday involve a notion of what the question is about.
- They differ in how structured the notion of aboutness is.
- They differ in how much the structure and form of the question determine what the question is about.

A detailed look at an end-to-end system: Watson

Watson

Most famous QA system: IBM's Watson ('DeepQA').

- Fame comes from playing Jeopardy, at which it was quite successful.
  `https://www.youtube.com/watch?v=WFR3lOm_xhE`
- PR 'how does it work' video:
  `https://www.youtube.com/watch?v=_Xcmh1LQB9I`.
  (Keep in mind: IBM is/was selling a product; Watson is now quite monetized.)
- Useful for outsiders because it is heavily documented in scientific papers.
- Many QA researchers will tell you: innovation is mainly architectural/engineering.

Watson

IBM Journal of Research and Development volume 56, 2012: Special issue on Watson.

`http://ieeexplore.ieee.org/xpl/tocresult.jsp?reload=true&isnumber=6177717`

Please email me if you want access to a dropbox folder collecting much of the QA literature I've been building on, including this material.

What is a Jeopardy question?

Watson didn't answer English questions!

- Jeopardy is a language game with its own convention about what a question is.
- I've never seen an analysis except informally by the Watson team.

What is a Jeopardy question?

Watson didn't answer English questions!

- Jeopardy is a language game with its own convention about what a question is.
- I've never seen an analysis except informally by the Watson team.
- Jeopardy game: answers have the form, 'wh- is X?'. Questions ('clues') are quite a bit more complicated.

What is a Jeopardy question?

Watson didn't answer English questions!

- Jeopardy is a language game with its own convention about what a question is.
- I've never seen an analysis except informally by the Watson team.
- Jeopardy game: answers have the form, 'wh- is X?'. Questions ('clues') are quite a bit more complicated.

Category: 18th century literature

This character studied medicine, "knowing it would be useful in long voyages". (`http://j-archive.com/`)

What is a Jeopardy question?

Watson didn't answer English questions!

- Jeopardy is a language game with its own convention about what a question is.
- I've never seen an analysis except informally by the Watson team.
- Jeopardy game: answers have the form, 'wh- is X?'. Questions ('clues') are quite a bit more complicated.

Category: 18th century literature

This character studied medicine, "knowing it would be useful in long voyages". (`http://j-archive.com/`)

Answer: 'who is Gulliver?'

What is a Jeopardy question?

Basic forms:

1. a sentence with a key proform in situ (termed the focus) that serves the role of an interrogative pronoun. Can be DP or D-sized. (e.g. 'he'/'she'/'this+NP'.)
2. a property-denoting DP/NP (by itself); the question is asking for a kind or individual that uniquely fills the property. Sometimes the category is needed.

Category: also a Batman Villain

Antarctic Krill-eater ()

What is a Jeopardy question?

Basic forms:

1. a sentence with a key proform in situ (termed the focus) that serves the role of an interrogative pronoun. Can be DP or D-sized. (e.g. 'he'/'she'/'this+NP'.)
2. a property-denoting DP/NP (by itself); the question is asking for a kind or individual that uniquely fills the property. Sometimes the category is needed.

Category: also a Batman Villain

Antarctic Krill-eater (answer: what(/who?) is The Penguin)

What is a Jeopardy question?

Basic forms:

1. a sentence with a key proform in situ (termed the focus) that serves the role of an interrogative pronoun. Can be DP or D-sized. (e.g. 'he'/'she'/'this+NP'.)
2. a property-denoting DP/NP (by itself); the question is asking for a kind or individual that uniquely fills the property. Sometimes the category is needed.

Category: also a Batman Villain

Antarctic Krill-eater (answer: what(/who?) is The Penguin)

3. underline for interrogative pronoun (e.g. 'In Billy Joel's "_____ Man" ')
4. Many other potential constraints on answers provided by category. (word-play, e.g. rhyming, number of letters, puns)

As a task, Jeopardy is great:

- Some (but not most) humans can do extremely well at it. (Also: PR value.)
- Large amount of historical data about questions, clues, categories, human performance.

As a task, Jeopardy is great:

- Some (but not most) humans can do extremely well at it. (Also: PR value.)
- Large amount of historical data about questions, clues, categories, human performance.
- Evaluation is easy: can we do better than a human?
- For the most part, it is the essence of factoid QA, so existing techniques apply.

The QA component of Watson

1. Analyze the question. (Many different kinds of analyses.)
2. Find potentially resolving information. (Try everything & the kitchen sink in parallel.)
3. Establish confidence values for all of this information.
4. Formulate answer based on information + confidence assessment.

The QA component of Watson ('DeepQA')

diagram from Ferrucci (2012):

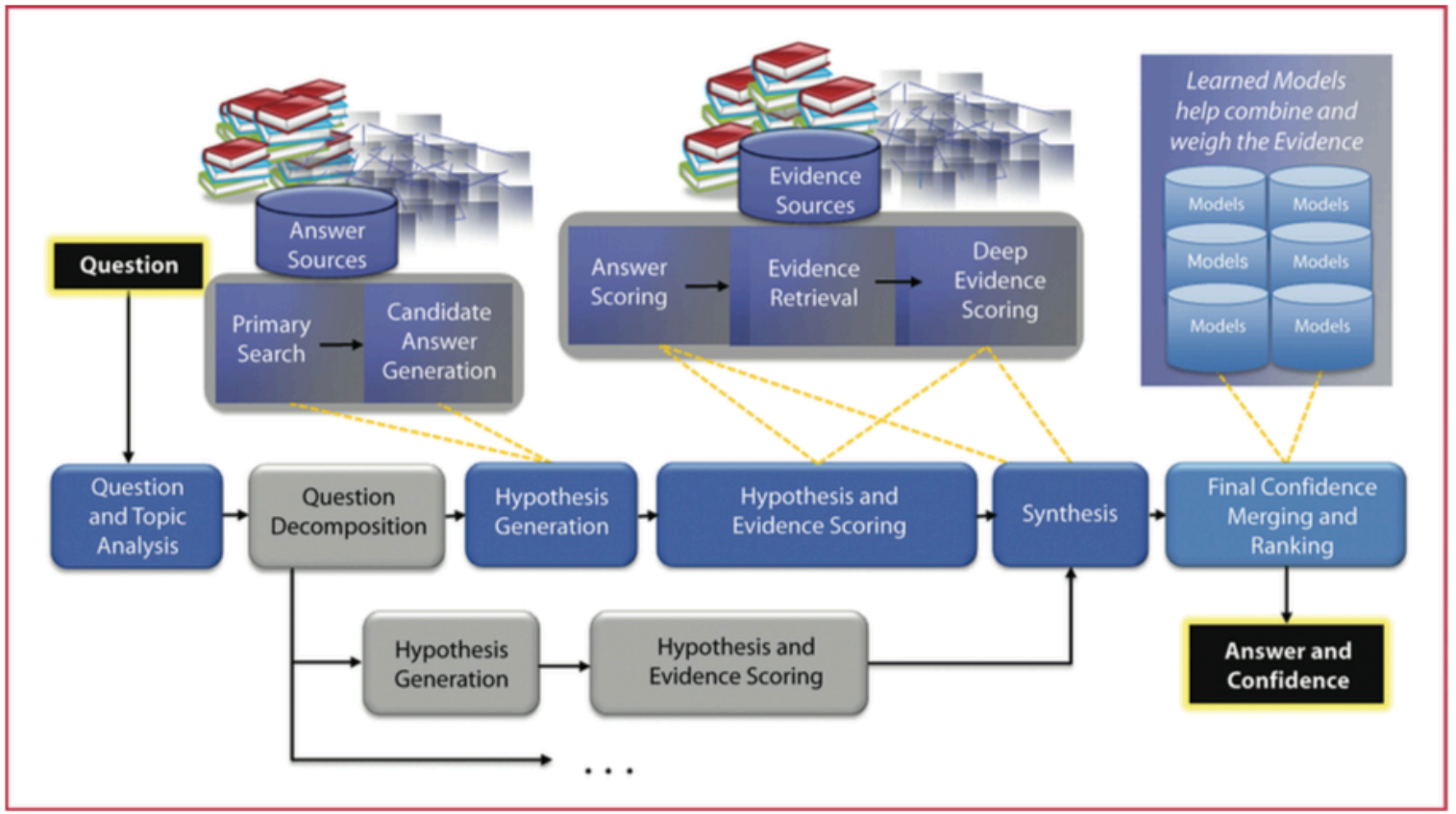


Figure 1

DeepQA architecture.

Question analysis

Example from question analysis paper

Category: poets and poetry

He was a bank clerk in the Yukon before he published ‘Songs of a Sourdough’ in 1907.

Key general-purpose steps

Throw state-of-the-art language technology at the question:

- Parsing (slot grammar parser; somewhat similar to dependency parsing).
- Named entity recognition
- Extract predicate-argument structure, e.g. basic semantics:

(2) publish(e1, he, “Songs of a Sourdough”)
in(e2, e1, 1907)
authorOf(focus, “Songs of a Sourdough”)
temporalLink(publish(...), 1907)
(...)

Relation extraction

How to map parses to relations? Large area of current research in computational semantics, under various names. (Semantic parsing, event extraction, event detection, etc.)

- Rule-based approach – Jeopardy-specific, as well as extremely general relations (`authorOf`, `bornWhere`). About 30 relations.
- Statistical approach: thousands of relations. Basic idea: build a topic model over data in a alarge knowledge-base to extract relations.
- DBPedia: semi-structured open knowledge-base over wikipedia `http://dbpedia.org/fct/facet.vsp?cmd=text&sid=29393`
- (Directly finding the answer in DBPedia via a semantic parse happened about 2% of the time. But that's independent of the relation extraction.)

Key jeopardy-specific steps

- Find the focus. (Heuristic)
- Find any terms that are properties of the focus. (‘Lexical Answer Terms’)
- Classify the question according to a jeopardy-specific scheme. (factoid, multiple-choice, definition, fill-in-the-blanks, etc.)
- Find special puzzle-related stuff that needs special handling, e.g. meta-linguistic properties of the answer.

Lexical answer term extraction

LATs seem to be a heuristic for key properties that restrict the focus.

- Can come from Jeopardy category or question.
- In question: focus-DP, but also from elsewhere in the question.

Category: here piggy, piggy, piggy

Many a mom has compared her kid's messy room to this kind of hog enclosure.

Lexical answer term extraction

LATs seem to be a heuristic for key properties that restrict the focus.

- Can come from Jeopardy category or question.
- In question: focus-DP, but also from elsewhere in the question.

Category: here piggy, piggy, piggy

Many a mom has compared her kid's messy room to this kind of hog enclosure.

Category: acresses' first films

Oklahoma!

Lexical answer term extraction

LATs seem to be a heuristic for key properties that restrict the focus.

- Can come from Jeopardy category or question.
- In question: focus-DP, but also from elsewhere in the question.

Category: here piggy, piggy, piggy

Many a mom has compared her kid's messy room to this kind of hog enclosure.

Category: acresses' first films

Oklahoma!

- Linguistic interpretation: heuristic model of contextual domain restriction, incorporating both the content of the question and the context.

Question classes

Watson's question categories are largely tailored to Jeopardy. Nonetheless they illustrate the challenge of coping with human flexibility in communication.

- My analysis: understanding these amounts to understanding the intent of the questioner in a Jeopardy context.

(Show Lally et al. (2012) table 4 here.)

Question Decomposition (Kalyanpur et al. 2012)

(3) parallel decomposition: This company with origins dating back to 1876 became the first U.S. company to have 1 million stockholders in 1951.

(4) nested decomposition: A controversial 1979 war film was based on a 1902 work by this author.

(5) (category: fictional animals) The name of this character, introduced in 1894, comes from the Hindi for ‘bear’.

The QA interpretation:

- Can be decomposed into multiple questions.
- A fair amount of work on this problem goin back to TREC.
- Linguistic interpretation (at least for much of this data): need a general account of projection.

Question analysis summary

All of this information is available to downstream components.

- Mix of heuristic and linguistically motivated analysis, including recognizable semantics.
- Heuristics are targeted at pragmatic issues: domain restriction, intent of questioner.
- General theme of Watson: need to be able to flexibly integrate a lot of information in parallel.
- Lots more to say about how these components are trained and evaluated; see Lally et al. (2012)

Search / hypothesis generation

Basic idea of hypothesis generation

Given a question analysis, look for answers in every way you can, then integrate the results later. Six main search mechanisms to find likely information sources:

- Four unstructured search components (basically like using a search engine).
- Two structured search components.

Converting questions to queries

Category: movie-“ing”

Robert Redford and Paul Newman starred in this depression-era grifter flick.

Converting questions to queries

Category: movie-“ing”

Robert Redford and Paul Newman starred in this depression-era grifter flick. (Answer: the Sting)

Relation structure:

(6) actorIn(Robert Redford; flick : focus), actorIn(Paul Newman; flick : focus)

Converting questions to queries

Category: movie-“ing”

Robert Redford and Paul Newman starred in this depression-era grifter flick. (Answer: the Sting)

Relation structure:

(6) actorIn(Robert Redford; flick : focus), actorIn(Paul Newman; flick : focus)

Full query (weights “empirically determined” determined by relation structure):

(7) (2.0 ‘Robert Redford’) (2.0 ‘Paul Newman’) star depression era grifter (1.5 flick)

LAT-only query:

(8) depression era grifter flick

Unstructured passage search

Not very interesting: use queries in various ways together with an open source search engine (Indra) to find relevant passages.

- Note that underlying this is the same basic topic modeling idea we talked about earlier.

Structured search

Use extracted relations to search structured data, query-style (DBPedia, PRISMATIC).

- Summary: very good, in a very small percentage of Jeopardy questions.

Candidate answer generation

Next problem: given some unstructured passage, how can you identify candidate answers in that passage?

- Answer: use lots of heuristics.

Confidence

Confidence assessment overview

Given some set of candidate answers from very heterogenous sources, how to assess confidence in answers?

- Different from most research QA systems, which rely on just one (type of) information source.
- Sometimes seen as a key reason for Watson's success.

Confidence assessment and supporting evidence

‘Supporting Evidence Retrieval’: Given some search query and a candidate answer:

- Rerun the search with the candidate answer integrated into the query.
- Score resulting passages on how well they match the original question. (terms, syntax, etc)

Confidence assessment and supporting evidence

‘Supporting Evidence Retrieval’: Given some search query and a candidate answer:

- Rerun the search with the candidate answer integrated into the query.
- Score resulting passages on how well they match the original question. (terms, syntax, etc)
- Provides some assessment of how relevant the candidate answer is to the question, independent of the initial search mechanism.
- Reverses the usual linguistics way of thinking about it (given some question, what answers would be relevant).

An example with SER involving logical representations

Silver Springs in northern Florida is one of the state's largest water-filled one of these holes.

An example with SER involving logical representations

Silver Springs in northern Florida is one of the state's largest water-filled one of these holes.

Watson without this module selects 'Ocala' as an answer (wrong).

An example with SER involving logical representations

Silver Springs in northern Florida is one of the state's largest water-filled one of these holes.

Watson without this module selects 'Ocala' as an answer (wrong).

Supporting passage that gets the correct answer:

Devil's Hole is a large water-filled sinkhole close to the south-eastern corner of Harrington Sound, Bermuda.

Adjusting confidence via answer typing

Lexical Answer Types are used to constrain the queries, but also to adjust confidence in candidate answers:

- Convert LATs to types in an ontology (various).
- Use lexical resources to type the candidate answer (WordNet, DBPedia, etc)
- Use degree of match as a (soft) weight on candidate answers.

Consolidating candidate answers

Really hard problem. Lots of complicated, iterative machine learning here.

- Attempt to merge candidate answers that refer to the same entity.
- Attempt to merge evidence for different candidate answers.
- Various manipulations that are specific to Jeopardy (treat some categories differently, that kind of thing)

Formulating an answer

Formulating an answer

Watson does not really have a separate formulation step.

- Initial candidate answer depends on source.
  Knowledge-base, just take node.
- If source is passage, use heuristic techniques to identify the candidate answer.
- In consolidation step, attempt to convert candidate answers to canonical form (e.g. JFK → John F Kennedy).
- Special question types may have their own heuristics.

Summary

(diagram from Ferrucci (2012))

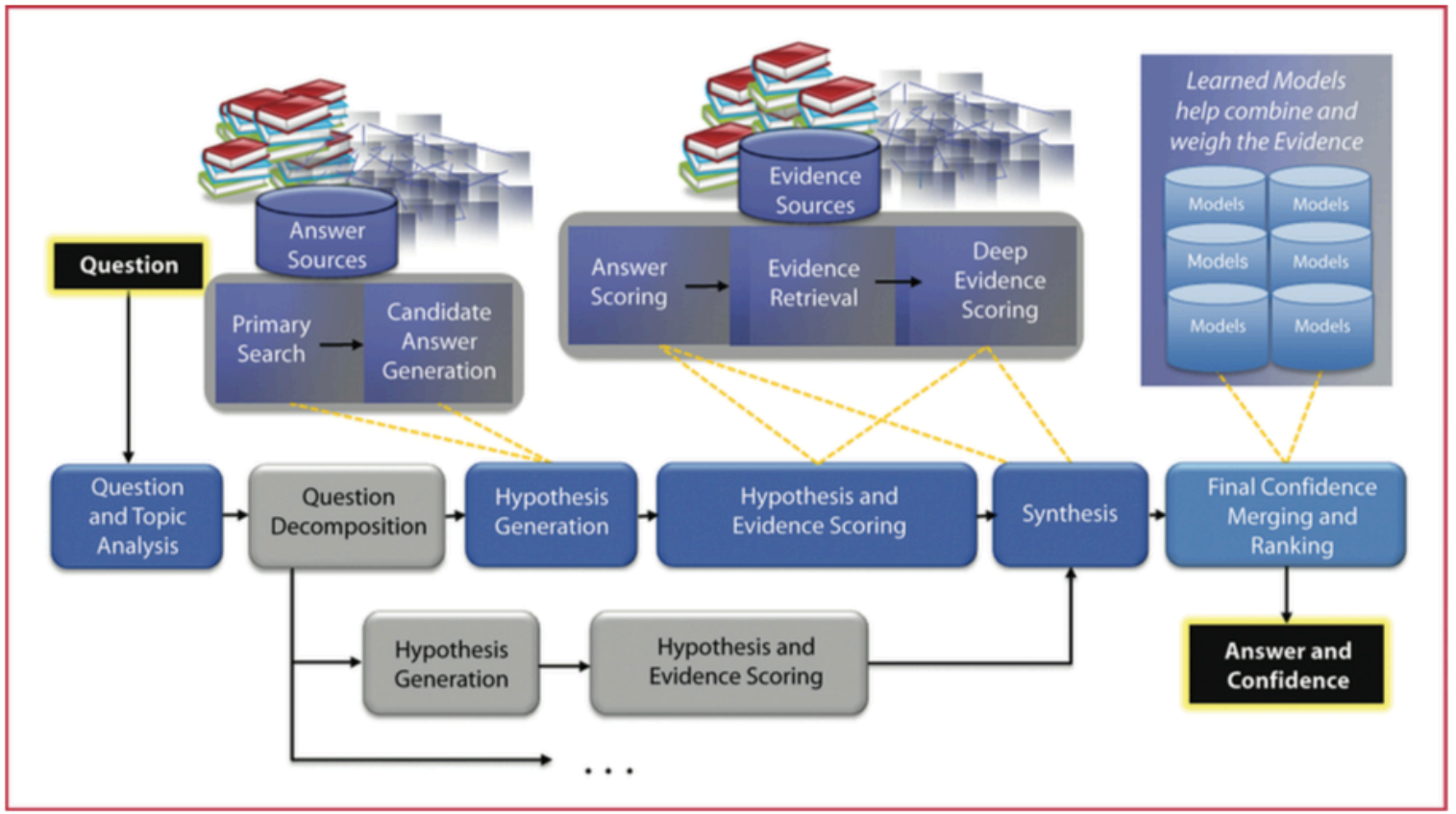


Figure 1

DeepQA architecture.

Bibliography I

Ferrucci, D. A. 2012. Introduction to “this is Watson”. *IBM Journal of Research and Development* 56.

Kalyanpur, A., S. Patwardhan, B. K. Boguraev, A. Lally & J. Chu-Carroll. 2012. Fact-based question decomposition in DeepQA. *IBM Journal of Research and Development* 56.

Lally, A., J. M. Prager, M. C. McCord, B. K. Boguraev, S. Patwardhan, J. Fan, P. Fodor & J. Chu-Carroll. 2012. Question analysis: how Watson reads a clue. *IBM Journal of Research and Development* 56.

Lewis, David. 1988. Relevant implication. *Theoria* 54. 161–237.

Moldovan, Dan & Mihai Surdeanu. 2003. On the role of information retrieval and information extraction in question answering systems. In M. T. Pazienza (ed.), *Information extraction in the web era*, 129–147. Springer.

Bibliography II

Rawlins, Kyle. 2013. About ‘about’. In *Proceedings of SALT 2013*, .

Yablo, Stephen. forthcoming. *Aboutness*. Princeton University Press.

Yao, Xuchen & Benjamin Van Durme. 2014. Information extraction over structured data: question answering with Freebase. In *Proceedings of ACL 52*, 956–966.