Watson

* QA technology can help professionals more precisely answer questions

# Metric

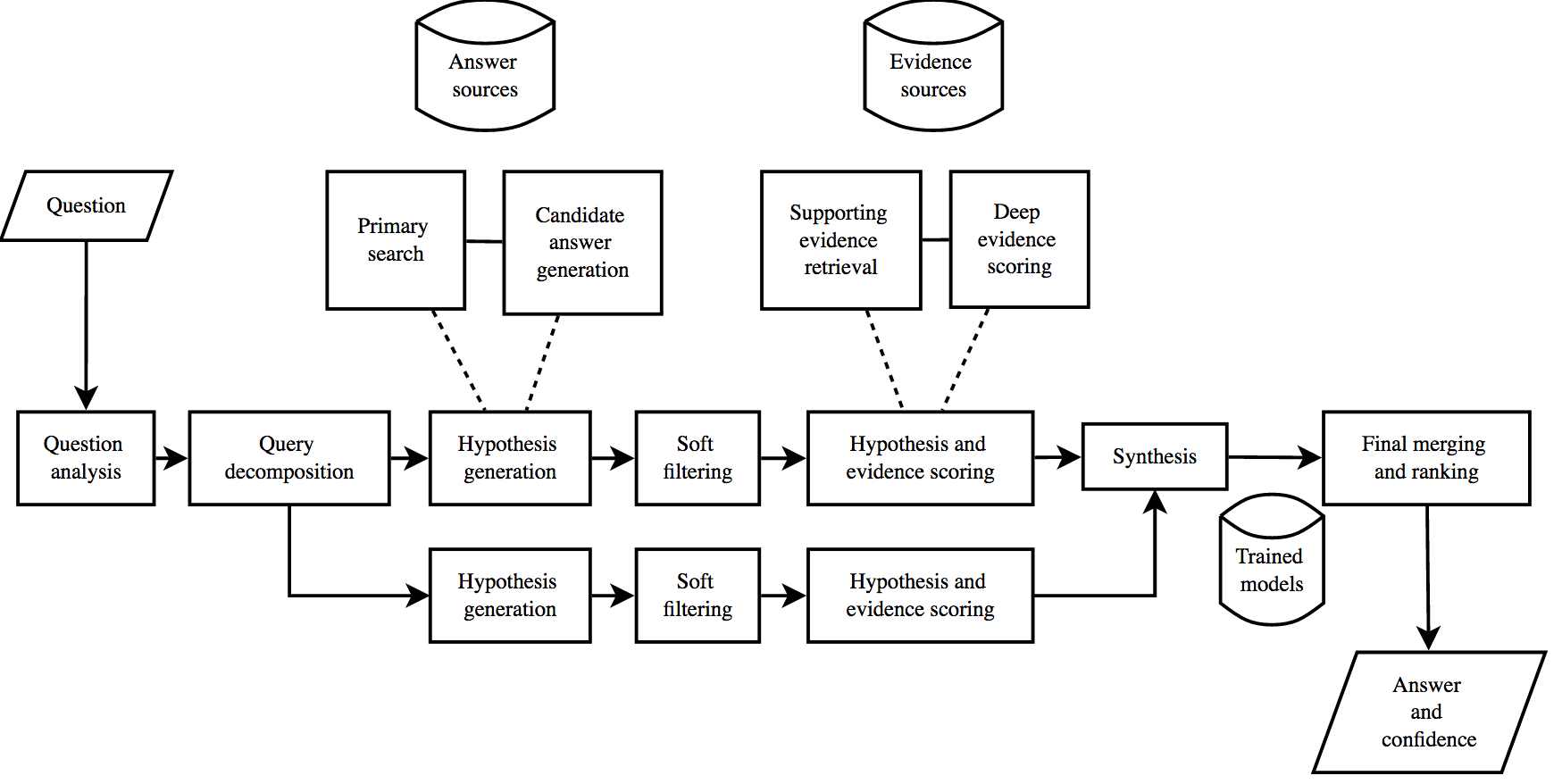
* Goal = measure correctness and confidence using “precision” and “percent answered”
  + Precision = percent of questions correct out of the ones that it answered
  + Percent answered = percentage of questions answered (correctly or incorrectly)
  + Choose which questions to answer based on confidence score threshold
    - Higher threshold = less percent answered, higher precision
    - Lower threshold = more percent answered, less precision
    - Change threshold to increase precision of a system with constant accuracy (accuracy = precision when percent answered = 100)
* Human champions in Jeopardy typically had 50% answered, with 80-100% accuracy
  + Ken Jennings had 62% answered, with 92% accuracy

# Baseline Performance

* IBM’s previous QA system = Practical Intelligent Question Answering Technology (PIQUANT) was part of the Text Retrieval Conference (TREC) QQ systems
  + TREC allows use of web, gives 500 questions, and has a time limit of 1 week, and has simpler questions than Jeopardy
  + Has 33% accuracy in TREC
  + PIQANT did terribly (of 5% of clues PIQUANT was most confident in it had 47% precision, but had mostly 13% precision)
* OpenEphyra’s open source QA framework at CMU answered 45% of TREC questions correctly
  + For Jeopardy, OphenEphyra had < 15% accuracy
* Basic text search, and structured data search also performed relatively poorly
  + Text search actually performs has better precision as % answered increases!
    - Tapers off at 30%
  + Structured search (database with known entities) has very high precision for small % answered, but quickly drops off to ~10% when asked to answer more questions

# DeepQA Approach

* Despite overhauling PIQUANT with many published algorithms, Jeopardy performance change was minimal
  + OAQA was initiated to get researchers in community to replicate and reuse research results



* Principles in DeepQA =
  + Massive parallelism – exploit massive parallelism in consideration of multiple interpretations and hypotheses
  + Many experts – use wide range of analytics
  + Pervasive confidence estimation – no component commits to answer; all components use confidences
  + Integrate shallow and deep knowledge: balance strict + shallow semantics

## Content Acquisition

* Basically – see previous jeopardy question types, and try to gather info from encyclopedias, dictionaries, thesauri, news articles, etc
* In order to have documents w/o using web, do the following ahead of time:
  + Get seed documents and retrieve related documents from web
  + Extract self contained text nuggets from related web documents
  + Score them based on informativeness to original seed document
  + Merge most informative nuggets into expanded corpus
* Also use DbPedia, Wordnet, and Yago

## Question Analysis

* Various techniques to break down question:
  + Shallows parse, dee parse (McCord 1990)
  + Logical forms
  + Semantic role labels,
  + Coreference,
  + Relations
  + Named entities
* Question Classification:
  + Task of identifying question types or parts of questions that require special processing
    - i.e. puzzle, math, definition, puns, subclues
* Focus and LAT detection:
  + LAT – lexical answer type; word in the clue that indicates the type of answer, independent of assigning semantics to the word, eg:
  + " Invented in the 1500s to speed up the game, this maneuver involves two pieces of the same color.
    - LAT is “maneuver”
  + Focus = part of question that if replaced by the answer makes the question a stand-alone statement (this seems more jeopardy specific)
* Relation Detection:
  + Subject-verb-object predicates or semantic relationships between entities
  + Watson uses relation detection (from focus + LAT determination), and can use it to query a triple store
  + But, due to breadth of relations in Jeopardy + huge number of ways they can expressed, Watson can only “look up” < 2% of clues
  + Even the 50 most frequently occurring freebase relations occur at most 25%
* Decomposition:
  + Rule-based deep parsing and statistical classification to recognize whether questions should be decomposed and to determine how best to break into subquestions (sounds like sempre!)
  + Solves parallel decomposable questions through application of end to end QA system on each subclue and synthesizes final answers
* Hypothesis Generation:
  + Takes the results of question analysis and searches system sources + extracts answer-sized snippets from search results
  + Each candidate answer is a hypothesis with some degree of confidence
  + Called Primary Search:
    - goal = find as much possible answer-bearing content as possible based on results of question analysis (high recall)
    - idea = use other deeper content analytics to extract answers + score the evidence
      * deepqa achieved 85% recall for top 250 candidates (system generates correct answer as a candidate answer for 85% of questions somewhere within top 250 ranked candidates)
    - multiple search techniques + multiple search queries for single question
      * text search engines: Indri, Lucene
      * Document search, passage search
      * Knowledge base search (SPARQL on triple stores)
      * Some LAT’s are “closed” (i.e. limited number of possible solutions, i.e. “country”, or “U.S. President”)
  + Search results fed into Candidate Answer Generation:
    - Techniques appropriate to kind of search results are applied to generate candidate answers
    - Eg: title of a document, passage text analysis, named entity detection, triple stores
    - If correct answer isn’t generated at this stage, no hope of answering question; so favors recall over precision
    - Filter out correct result later
* Soft Filtering
  + Lightweight scoring algorithm applied to larger set of initial candidates to prune them down to smaller set of candidates before more intensive scoring components
  + Many small lightweight analysis scores combined into a single soft filtering score
  + If it passes threshold, it continues onto hypothesis + evidence scoring
  + Scoring model + filtering threshold = based on machine learning over training data
* Hypothesis and Evidence Scoring
  + Candidate answers undergo “rigorous evaluation process”; gathering additional supporting evidence + deep scoring analytics
  + Evidence Retrieval:
    - Many evidence gathering techniques used:
      * Passage search where the candidate answer is added as required term to the primary search query derived from the question
        + Gets passages where candidate answer is used in the context of original question terms
        + Supporting evidence may also come from triple stores
    - Evidence is sent to deep evidence scoring components
  + Scoring:
    - Where most of deep content analysis is performed:
      * Determine degree of certainty that retrieved evidence supports the candidate answers
    - DeepQA has more than 50 scoring components; each has few restrictions on score semantics
      * Formal probabilities, counts
      * Counts of categorical features,
      * Degree of match between a passage’s predicate-argument structure and question
      * Passage source reliability
      * Geospatial location
      * Temporal relationships
      * Taxonomic classification
      * Lexical and semantic relations candidate is known to participate in
      * Candidate’s correlation with question terms
      * Candidate’s popularity, aliases, etc
    - EG:
      * Counts of idf weighted terms in common between question and passage
      * Smith waterman sequence matching algorithm measures lengths of longest similar subsequences between the question and passage
      * Alignment of logical forms of the question and passage (where nodes = terms in text, edges = grammatical relationships), Hermjakob, Hovy, and Lin [2000]; Moldovan et al. [2003]
      * Deep semantic relationships (Lenat 1995, Paritosh and Forbus [2005])
      * Use knowledge in triple stores (for taxonomies, geospatial, and temporal reasoning)
    - Deepqa combines all of these scores into an evidence profile
    - Evidence profile groups individual features into aggregate evidence dimensions that provide a more intuitive view of the feature group
      * Different dimensions might be weighed higher based on the question itself
* Final Merging
  + Final piece of puzzle = get precise answer by evaluating 100s of hypotheses based on 100,000s scores + get single best hypothesis and estimate its confidence
  + Some answers may be different surface forms of the same answer
    - Merge answer scores before ranking and confidence estimation
    - Uses matching, normalization, and coreference resolution algorithms
    - (maybe use freebase entity list here?)