Decision Making in IBM Watson™ Question Answering

Dr. J. William Murdock
IBM Watson Research Center

IBM Watson

- **IBM Watson** is an automated question answering system.
- It competed against Jeopardy!'s two all-time greatest champions.
- This match appeared on television in February of 2011.
- Watson won the match, outscoring both opponents combined.

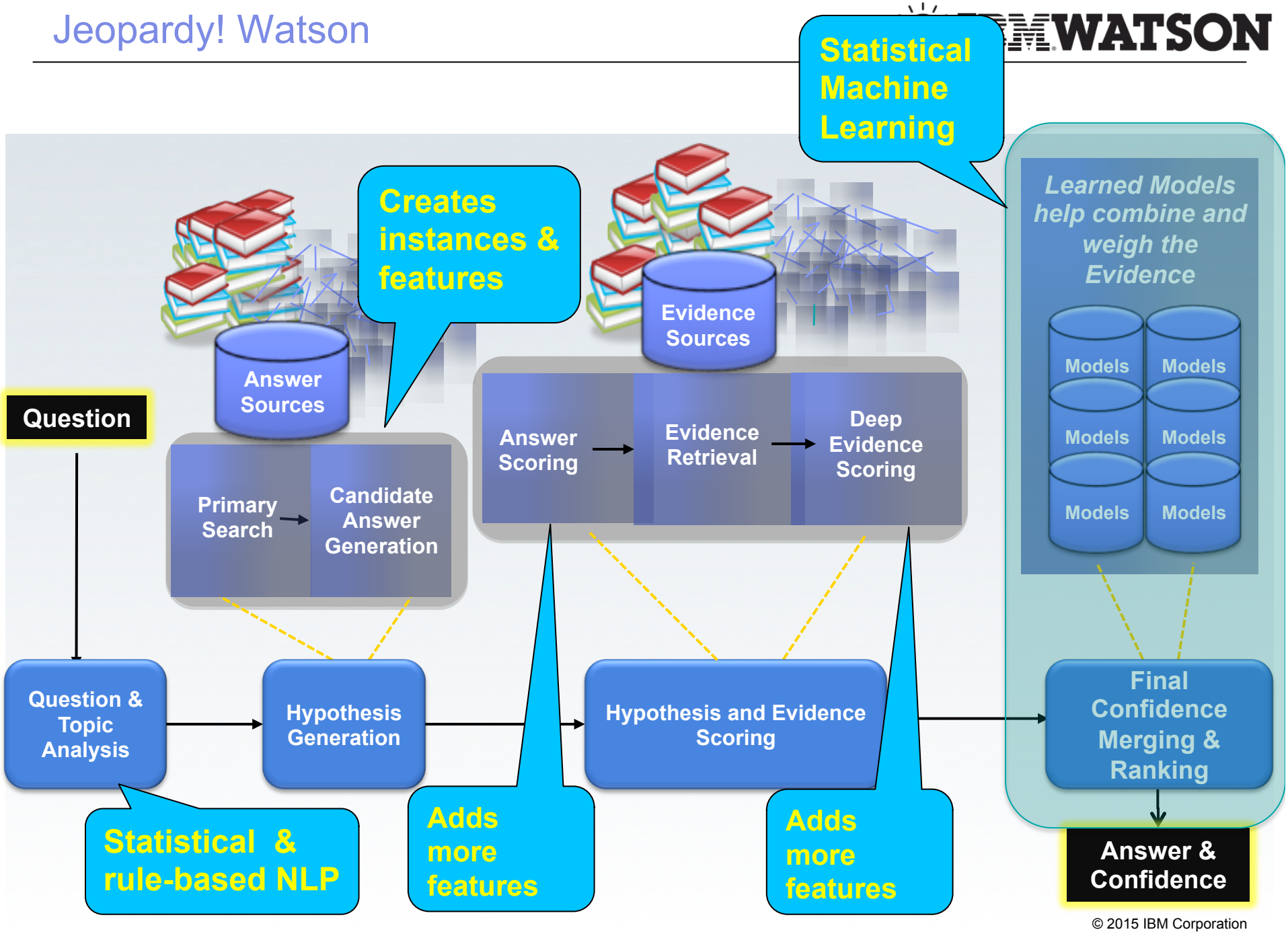
More recent work on IBM Watson focuses on business applications such as medicine and customer service.

Decision Making in Question Answering

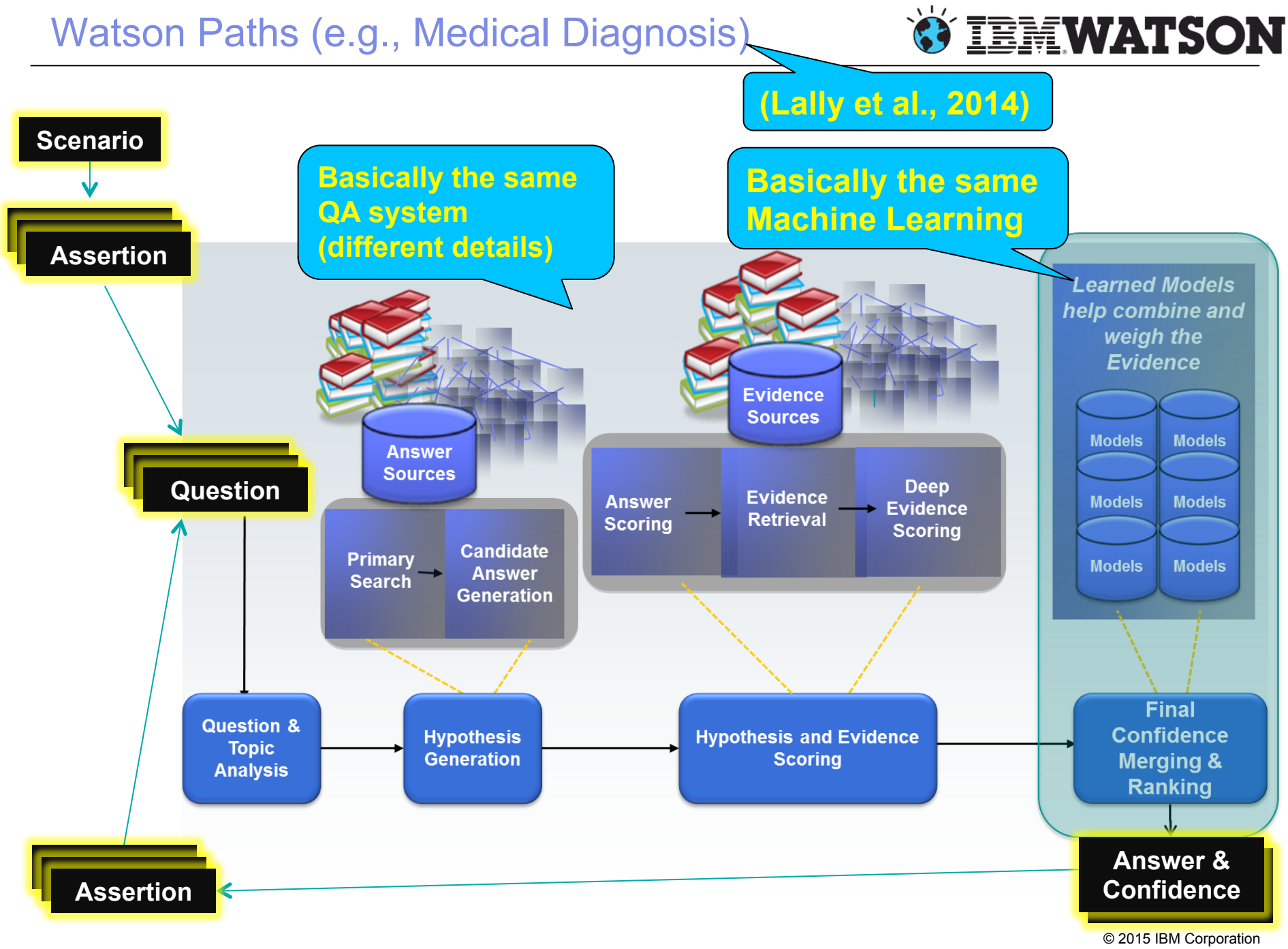
IBM WATSON

- **Choosing answers to questions!**
  - IBM Watson generates many candidate answers
  - For each answer, how confident are we that the answer is right?
- Deciding whether to answer
  - Based on how confident we are that the answer is right
  - Based on cost/benefit of right answers and wrong answers
- Deciding how many answers to provide
- Deciding whether to hedge

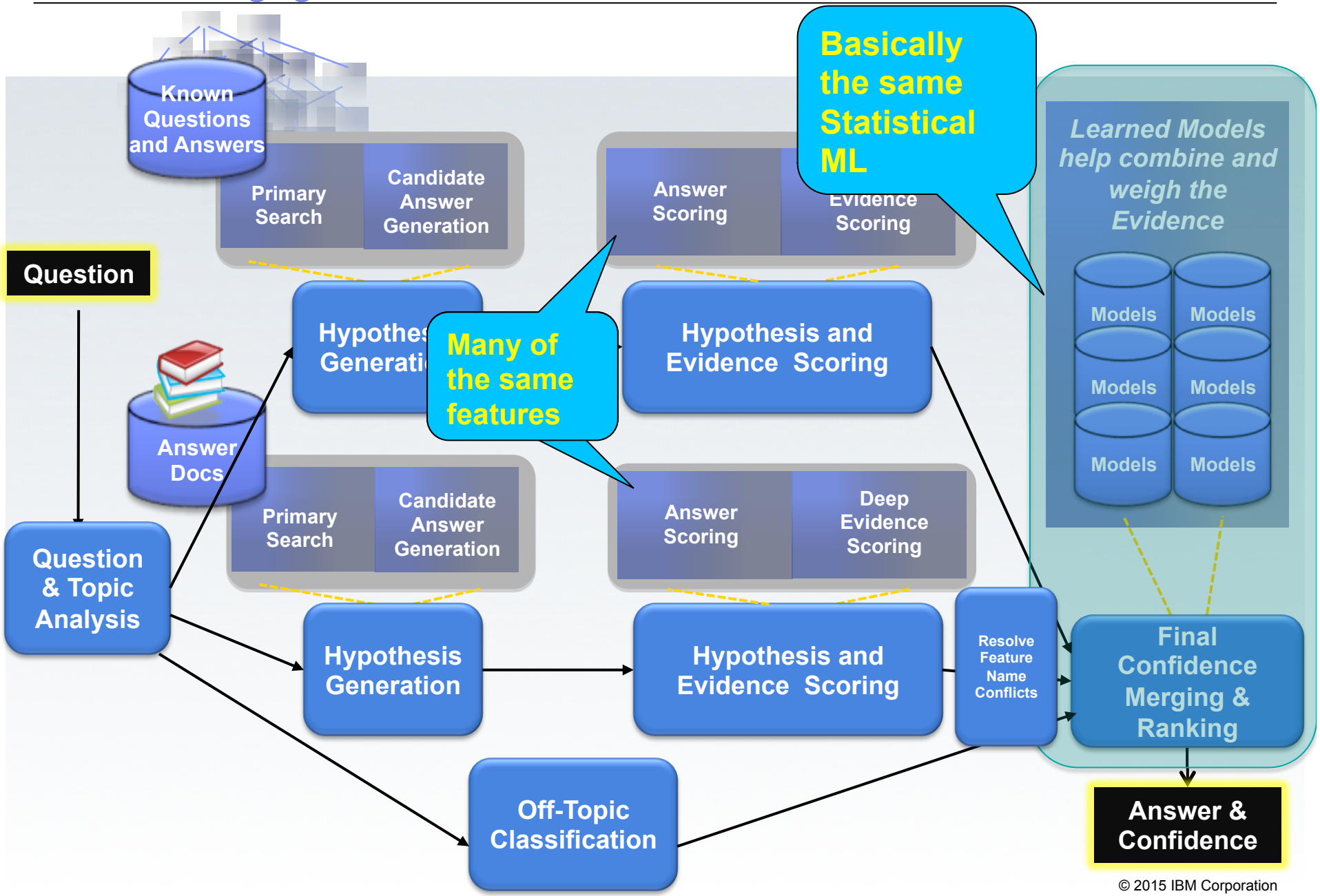
Jeopardy! Watson



Watson Paths (e.g., Medical Diagnosis)

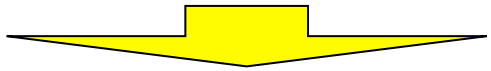


Watson Engagement Advisor



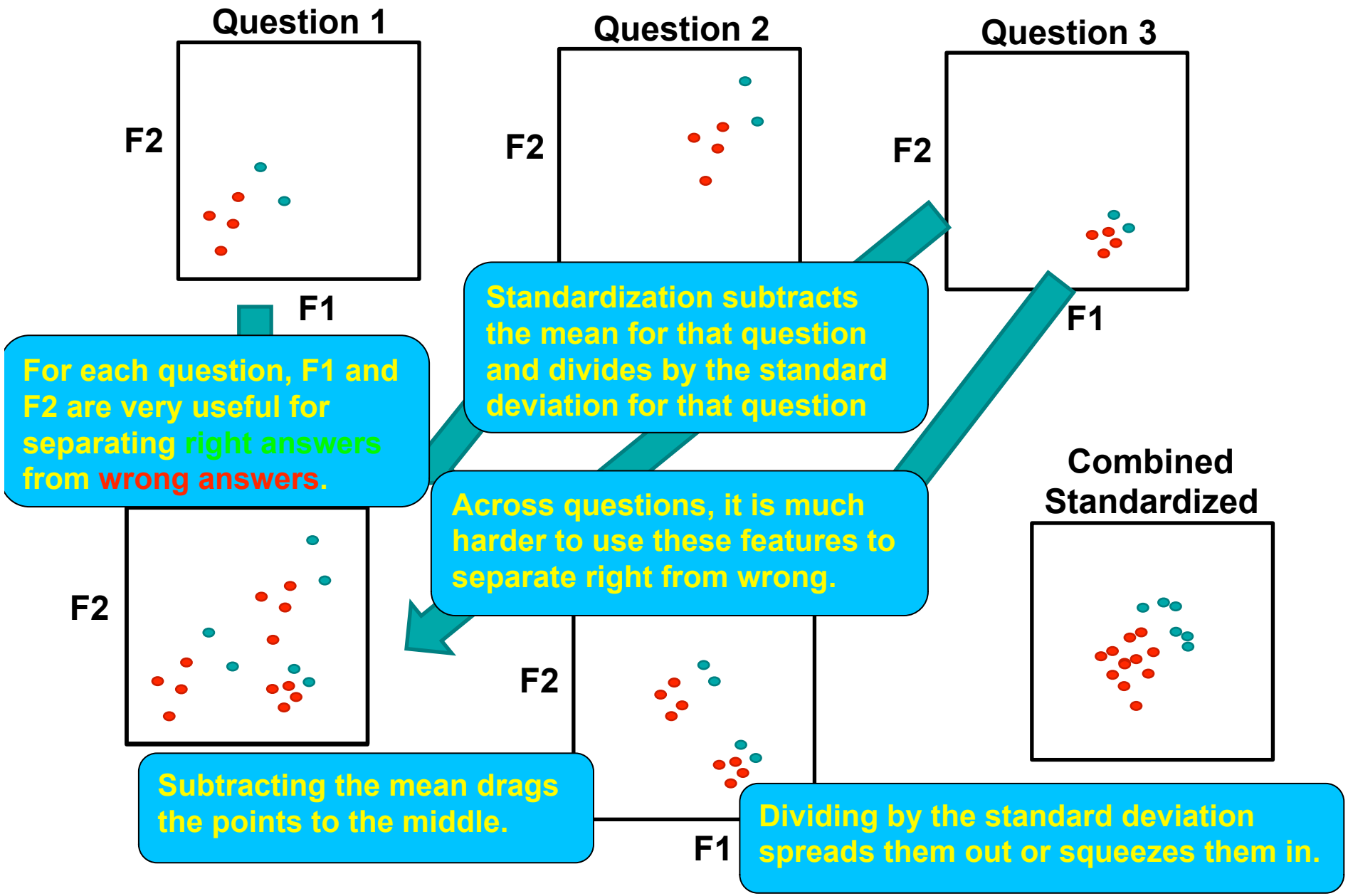
Final Confidence Merging & Ranking

- Start with:
  - Answers in isolation, with feature values
  - Answers embedded in evidence, with feature values
- Configured for each feature:
  - What to do when it is missing (typically if F is missing, then set F=0 and set F-Missing=1)
  - Standardization (see next slide)
  - How to merge across instances? (Max? Min? Sum? Decaying sum?)
- Combine “equivalent” answers, merging feature values
  - “frog”:f1=3,f2=4,f3=1
  - “order Anura”:f3=3
  - “frog”:“Frogs like ponds”:f4=0.8,f5=4.5
  - “frog”:“I saw a frog in a pond”:f4=0.6,f5=2.5



  - “frog”: f1=3,f2=4,f3=1,f4=1.1,f5=4.5
- Run ML
  - At training time, record features and class label (right or wrong) for training a model
  - At run time, apply the model to the features to compute a confidence score

Standardization



Phases and Routes

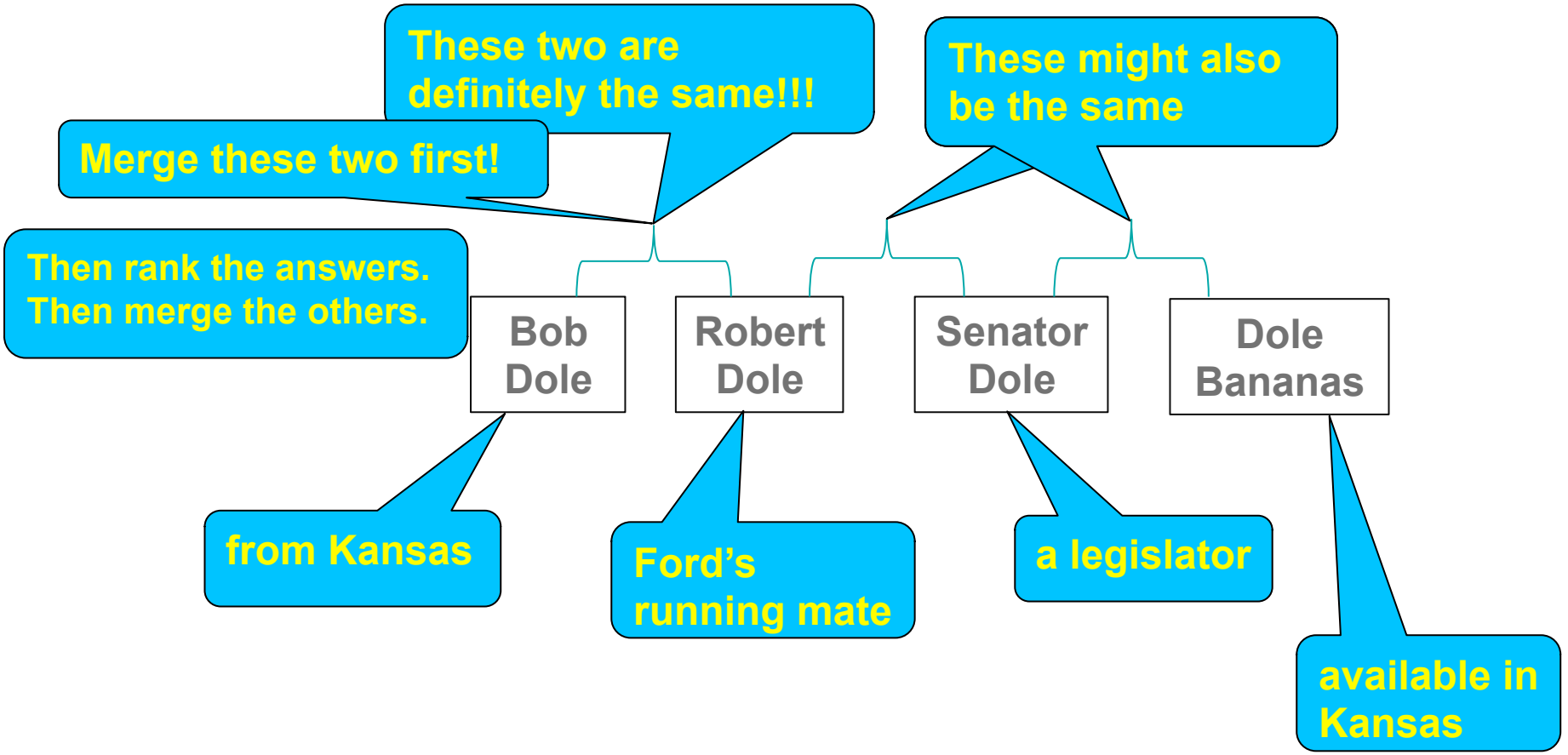
- Merging in ranking occurs in multiple *phases*
- All answers to all questions are processed in all phases
- Phases are used for sequential learning operations: cases where the outputs of earlier learning should influence later learning.
- (Example on next slide)

- Within each phase are alternative *routes*
- Within a phase, all answers to a single question are processed by only one route
- Routes are generally used for special kinds of questions
  - In Jeopardy! this includes puzzles, multiple-choice questions, etc.
  - In business applications in can include factoid, definition, why, how-to, and many more.

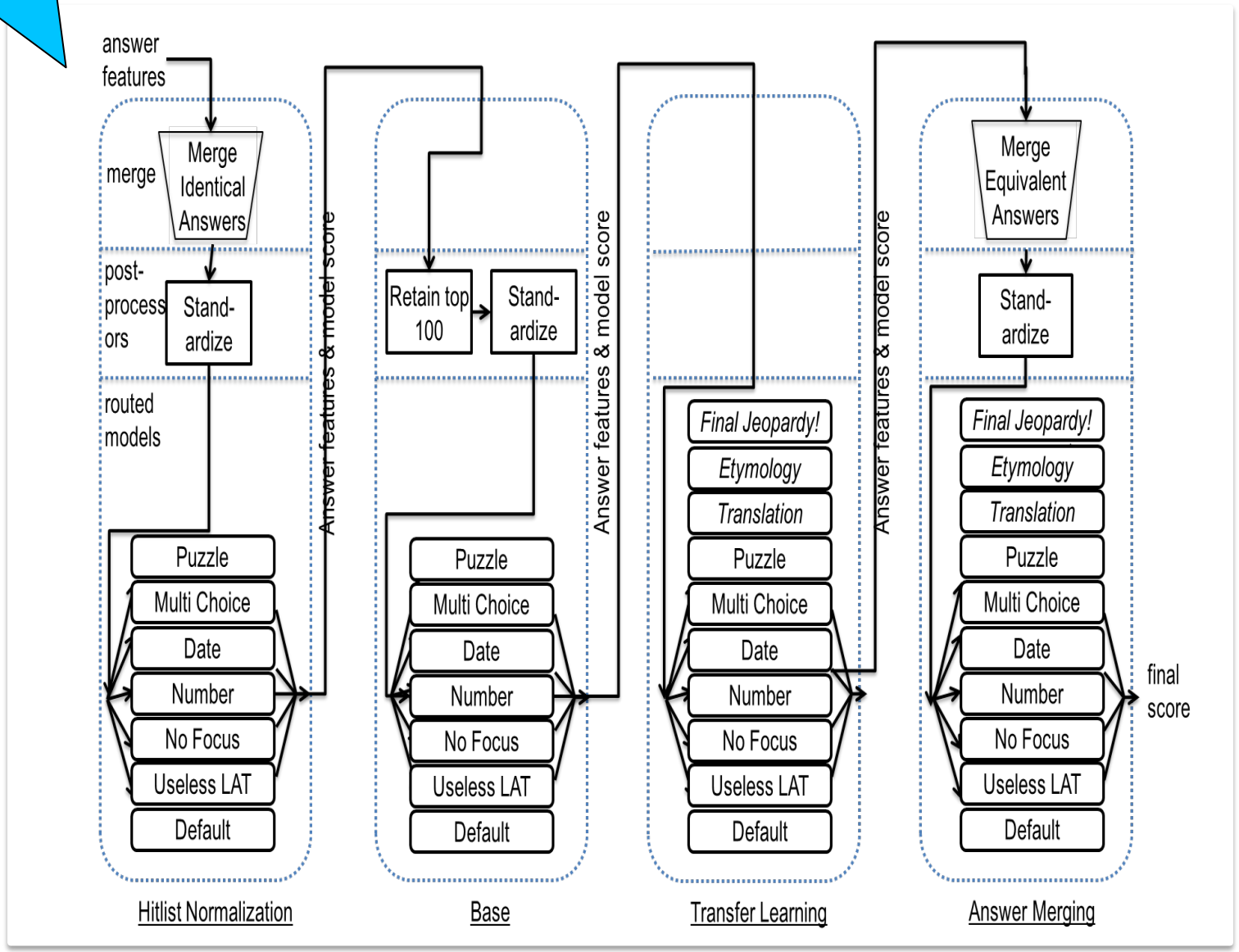
Phases: Answer Merging (Made-Up) Example

IBM WATSON

1976: This Kansas legislator was Gerald Ford's running mate.



(Gondek et al., 2011)



Learning to rank?

IBM WATSON

- Jeopardy! Watson (1.0) and all the other configurations described so far rank answers one at a time using logistic regression.
  - Learning is instance based.
  - One instance is a single answer to a single question.
- *Should* be possible for a system to do a better job ranking answers by looking at the answers as a group and learning to rank them instead of just scoring them one at a time in isolation.
- Learning to rank is very successful in other applications
- We tried learning to rank for Jeopardy! Watson and found it was not more effective than multi-phase logistic regression.
  - We're not *really* scoring them one at a time in isolation: standardization and multiple phases are effective ways to let answers influence each other.
  - We design features and test them using logistic regression and discard the ones that are not effective. Result: features are well suited to logistic regression.
- However, the field of learning to rank has continued to progress.
- **Stay tuned!!!**

Decision Making in Question Answering

IBM WATSON

- Choosing answers to questions!
  - IBM Watson generates many candidate answers
  - For each answer, how confident are we that the answer is right?
- Deciding whether to answer
  - Based on how confident we are that the answer is right
  - Based on cost/benefit of right answers and wrong answers
- Deciding how many answers to provide
- Deciding whether to hedge

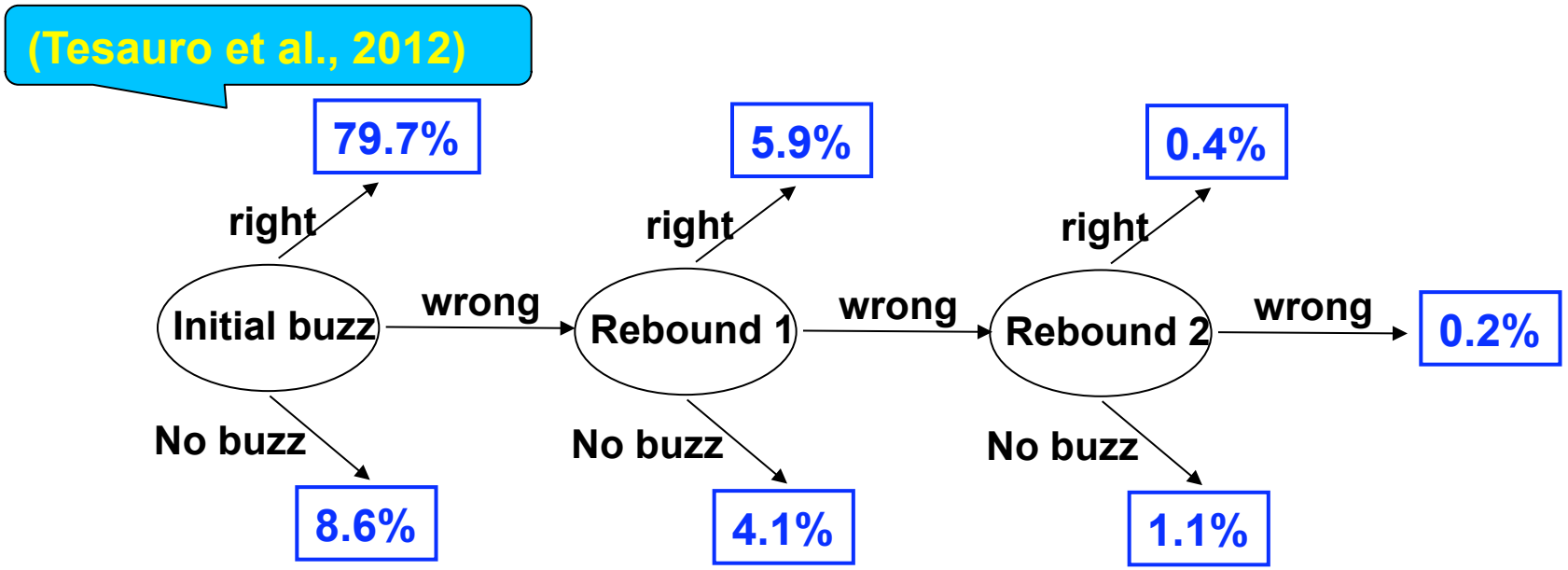
Deciding whether to answer

IBM WATSON

- In Jeopardy!, if you answer wrong, you lose money (hence the name).
- Very important to not answer when you don't know the answer.
- A big part of what motivated us to pursue Jeopardy!: Systems that know what they don't know.
- Lots of data in Jeopardy! to optimize win rate based on outcomes.
- Jeopardy! Watson (1.0) had extensive game playing capability for key Jeopardy! decisions like what category to select, how much to wager, and **whether to try to answer a question** (see upcoming slides)
  - The core Watson question answering capability is responsible for selecting an answer AND deciding how likely the answer is to be correct.
  - The Watson Jeopardy! playing application decides whether to answer using the confidence AND the game state (e.g., if Watson is way behind, it may take bigger risks).
- Similarly in business applications, core Watson QA ranks answers and assigns confidence scores; each application has its own logic for what to do with it.
  - For example, applications with very casual users may want to avoid wasting the users time with answers with a low probability of correctness
  - In contrast, applications used by highly motivated users may be more aggressive.

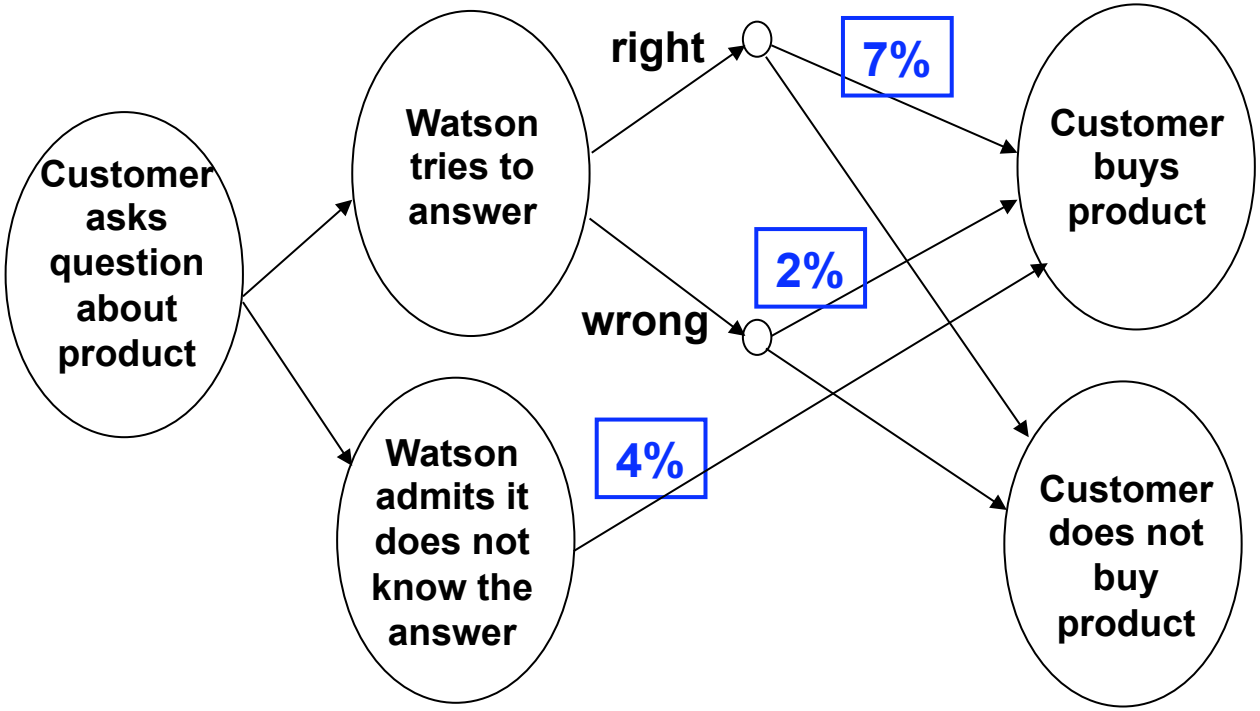
Stochastic Process Model of Jeopardy! Clues

- For Jeopardy! we had lots of data about human opponents behavior.
- This plus our own confidence in our own answer allows us to precisely model expected outcomes of answering or not answering.



Stochastic Process Model in Business Applications?

- We could do something similar in many business applications, e.g.:



- (customer is most likely to buy the product if Watson answers correctly but more likely to buy if Watson doesn't answer than it if it answers wrong)
- However, typically you don't really have data like that.
- Core Watson just provides answers + confidence, applications must decide.
- Mostly just have a pre-defined confidence threshold (subjective user experience).

Decision Making in Question Answering

IBM WATSON

- Choosing answers to questions!
  - IBM Watson generates many candidate answers
  - For each answer, how confident are we that the answer is right?
- Deciding whether to answer
  - Based on how confident we are that the answer is right
  - Based on cost/benefit of right answers and wrong answers
- Deciding how many answers to provide
- Deciding whether to hedge

Deciding how many answers + hedging

- In Jeopardy!, you can give only one answer.
- However, in real life you can show multiple answers to a user.
- Generally not as good as just providing the one correct answer
- However, sometimes IBM Watson has several answers that all have reasonably good scores and no one answer with a very good score.
- Show one answer when you are very confident of one answer, and show multiple answers you have several that all have similar confidences?

- A system can *hedge* an answer, by saying something like: “I don’t really know, but I think the answer might be…” before answering.
- It would do this for medium confidence answers: high enough to be worth showing a user but not high enough to
- Users may be more forgiving of wrong answers if they are hedged.

- Like deciding whether to answer, **these are addressed at the application level**. The core IBM Watson QA system provides many ranked answers with confidences, and the application decides what to do with that information.

References

- A Lally, S Bagchi, M A Barborak, D W Buchanan, J Chu-Carroll, D A Ferrucci, M R Glass, A Kalyanpur, E T Mueller, J W Murdock, S Patwardhan, J M Prager, C A Welty. WatsonPaths: Scenario-based Question Answering and Inference over Unstructured Information. Technical Report Research Report RC25489, IBM Research, 2014.
- D C Gondek, A Lally, A Kalyanpur, J W Murdock, P Duboue, L Zhang, Y Pan, Z M Qiu, C A Welty. A framework for merging and ranking of answers in DeepQA. IBM Journal of Research and Development 56(3/4), 14, 2012
- J W Murdock, G Tesauro. Statistical Approaches to Question Answering in Watson. Mathematics Awareness Month theme essay, Joint Policy Board for Mathematics (JPBM), 2012
- G Tesauro, D Gondek, J Lenchner, J Fan and J Prager. Simulation, Learning and Optimization Techniques in Watson's Game Strategies. IBM Journal of Research and Development 56(3/4), 14, 2012.
- D Ferrucci, E Brown, J Chu-Carroll, J Fan, D Gondek, AA Kalyanpur, A Lally, J W Murdock, E Nyberg, J Prager, N Schlaefer, and C Welty. Building Watson: An overview of the DeepQA project. AI Magazine 31(3), 59-79, American Association for Artificial Intelligence, 2010.