» A Hardware & Software Overview

Eli M. Dow <emdow@us.ibm.com:>

Overview:

» *Hardware*
» *Software*
» *Questions*

Early implementations of Watson ran on a single processor where it took 2 hours to answer a single question

Luckily, the DeepQA computation is embarrassing parallel

The rest of these slides will attempt to show how and why

Hardware

» *IBM Power750*
» *CPU*
» *Memory*
» *Networking*

This is where the magic happens...

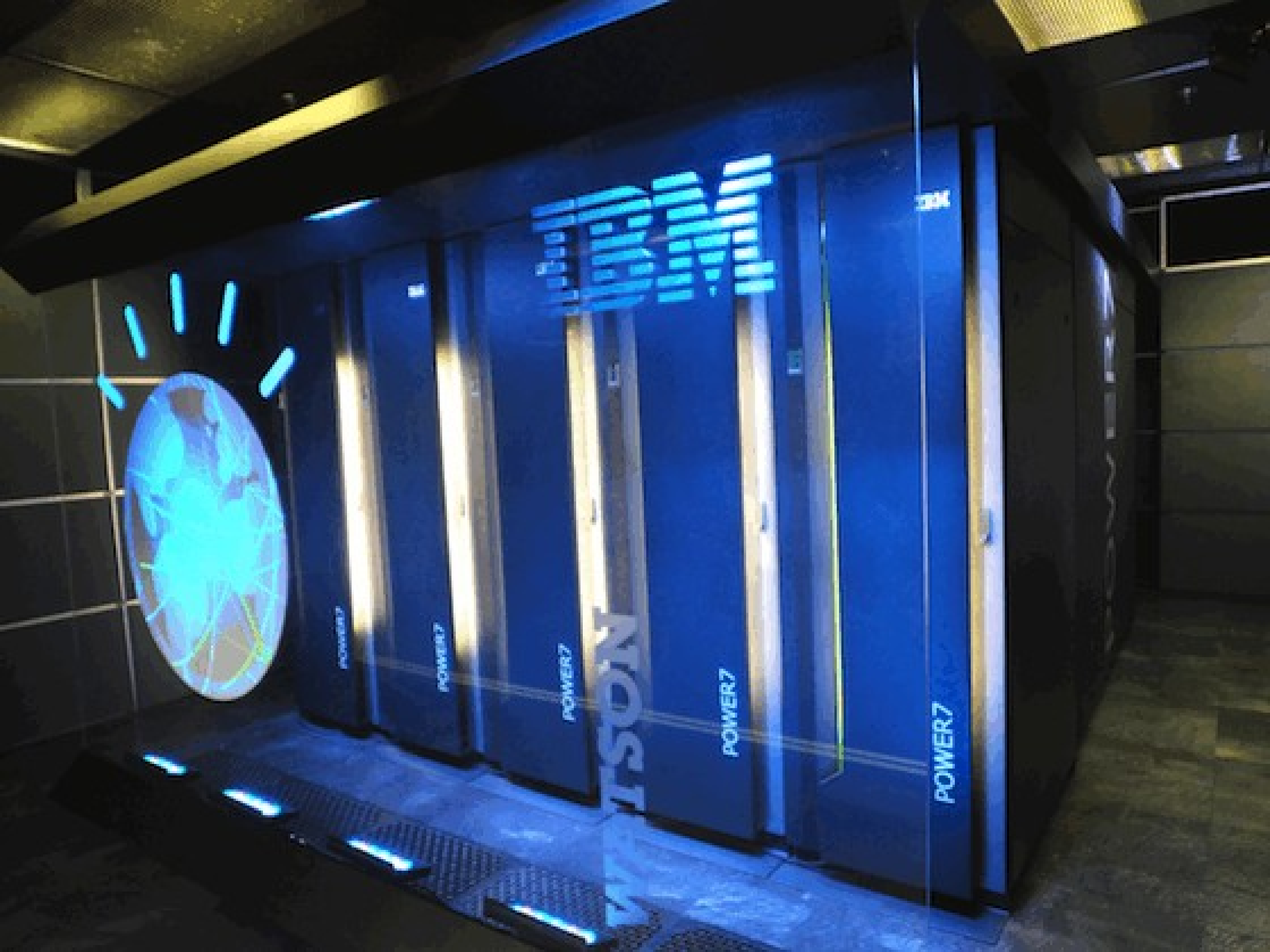
POWER 750: 1-4 sockets, With Max 32 Logical Cores

2,880 POWER7 Processor Cores

So Watson has 90 Power750 Servers

$$\frac{\text{2880 cores}}{\text{1 Watson}} \times \frac{\text{1 Server}}{\text{32 cores}} = \text{90 Servers}$$

10 Racks of IBM POWER 750 Servers


IBM
POWER7
POWER7
POWER7
POWER7
POWER7
WATSON

The Watson hardware is capable of around 80 TeraFLOPs

That is to say, 80,000,000,000,000 operations/second

Watson would likely be ~ 114th on the Top 500 Supercomputers list.

16 TB of RAM

Well equipped desktops have 0.007 TB Ram (8GB)

zEnterprise Business Class:
1.4 TB RAM
(248GB on a Model M10)

zEnterprise Enterprise Class
~3 TB RAM
(3,056GB on a Model M80)

1 TB could hold 17,000 hours of music.

To put that in perspective that's 708 days of non-stop listening to music (~2 years)

1 TB = 320,000 HD photos

Watson could store More than 5.1 Million

1 TB could hold 250 DVDs

1 TB could hold 1,000 copies of the Encyclopedia Britannica.

10 TB could hold the printed collection of the Library of Congress.

Watson only really stores about 1TB of actual text data (~1,000 x Encyclopedia Britannica).

Data was stored in RAM because the latency is too high when seeking on rotational disks.

“The network is the computer”

- John Burdette Gage

Networking required for Watson:

Watson is self-contained and not connected to the Internet.

But it has to move around a lot of data between compute nodes with ridiculously low latency...

1 x IBM J16E (EX8216) switch - populated with 15 x 10GbE line cards.

The J16E (EX8216) is massively scalable providing a 12.4 Tbps fabric and 2 billion packets-per-second (pps) line-rate performance.

Juniper Networks

All of the hardware is available for sale at your friendly international purveyor of business machines.

Estimated retail cost of similarly equipped IBM Power 750 server: $34,500 USD.
For 90 Servers → ~ $3 Million USD

Realistically, most organizations do not need 3 second response times → smaller configurations.

Software

» *Operating System*
» *MiddleWare*
» *Custom Software*

So lets talk about the operating system Watson uses...

This Operating System was originally developed by a Finnish student working alone in 1991

Windows

Watson has fallen asleep at the wheel. In fact, I totally forgot what I was doing and decided to let Jennings have this one. If no action is taken, please tell Trebek to go to commercial break while someone reboots me.

* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue _

Answer: Linux

Linux Enterprise Server 11

This relational database server runs on Windows, z/OS, AIX, Linux, and IBM i

Answer:

IBM DB2

Specifically:

VERSION HERE

IBM DeepQA software

Content Acquisition

The First step of DeepQA is content acquisition

The Goal is to identify relevant content source material to harvest possible answers from

Combination of manual and automatic steps involved:

- What kinds of questions will be asked?
- Characterize the application domain
- Analyzing example questions → primarily manual
- Domain analysis → automatic/statistical analysis helps

Highly Relevant Content

Given the kinds of questions and broad domain of the *Jeopardy* Challenge, the sources for Watson include a wide range of encyclopedias, dictionaries, thesauri, news articles, literary works, music databases, IMDB etc

Content Acquisition Continued

From a reasonable baseline corpus of text, we then want to automatically expand that textual information by adding other relevant, informative text.

Generally this is how it works:

1) Identify seed documents and retrieve related ones from the web
2) Extract self-contained text snippets from the related web docs
3) Score the snippets based on whether they are informative w/ respect to the orig seed doc
4) Merge the most informative snippets into the expanded corpus

This information comprises Watson's unstructured knowledge which is the bulk of the information used to answer questions

Content Acquisition Continued

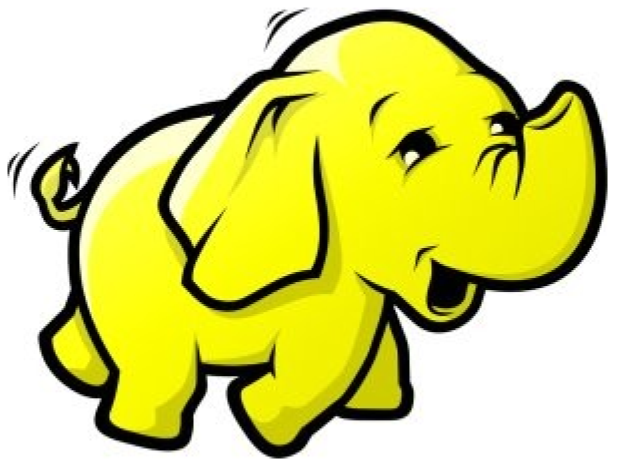
DeepQA leverages other kinds of semistructured and structured content as well

Another step in the content-acquisition process is to identify and collect these resources, which include databases, taxonomies, and existing ontologies

The live system itself uses this expanded corpus and does not have access to the web during play

Apache Hadoop

http://hadoop.apache.org/



Question Analysis

The first step in the run-time question-answering process is question analysis.

During question analysis the system attempts to understand what the question is asking and performs the initial analyses that determine how the question will be processed by the rest of the system

Well known areas of research covers some of this

- Relations
- Named Entities
- Logical forms
- Semantic labels

Question Analysis Continued

We will briefly talk about 3 of the question analysis methods that helped with jeopardy:

1) Question Classification
2) Lexical Answer Typing
3) Relation Detection

Question Analysis 1/3: Question Classification

Question classification - the task of identifying question types, or parts of questions, that require special processing

Question classification may identify:

- puzzle questions
- math questions
- definition questions
- etc

Include anything from single words with double meanings to entire clauses that have certain syntactic, semantic, or rhetorical purpose

Question Analysis 2/3: LAT Detection

A lexical answer type is a word or noun phrase in the question that specifies the type of the answer without trying to understand its semantics. It answers what kind of answer the question is looking for, not the answer itself.

LAT determination is important for confidence scoring:

Wrong type → low confidence. Right type → higher confidence.

DeepQA uses many independently developed answer-typing algorithms

Question Analysis 3/3: Relation Detection

Most questions contain relations, whether they are syntactic subject-verb-object predicates or semantic relationships between entities:

For example, in the question, “They’re the two states you could be reentering if you’re crossing Florida’s northern border,” we can detect the relation borders(Florida,?x,north).

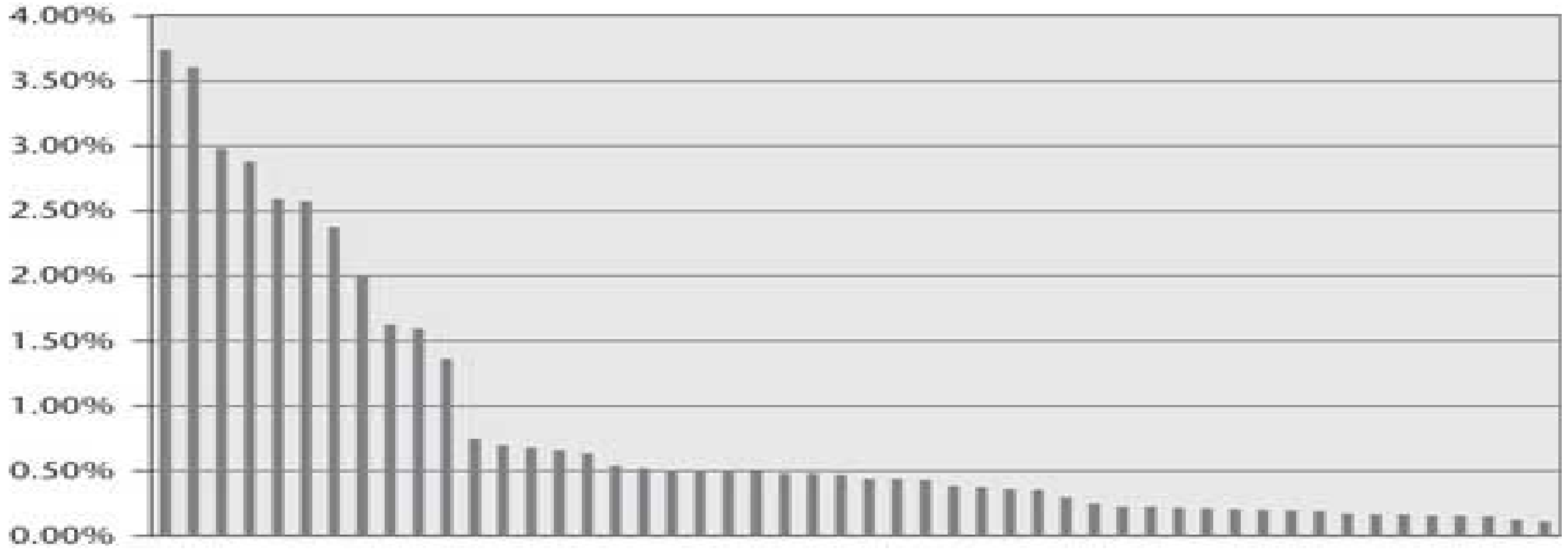
Watson uses relation detection throughout the QA process and can use detected relations to query a triple store and directly generate candidate answers

The breadth of relations in the *Jeopardy* domain and the variety of ways in which they are expressed means Watson effectively uses curated databases to “look up” answers in less than 2% of clues

Question Analysis 3/3: Relation Detection

For the DB2 experts in the room, a giant DB2 database wont work...

Watson's use of existing databases depends on the ability to analyze the question and detect the relations covered by the databases. In 20,000 *Jeopardy* questions IBM found the distribution of relations extremely flat:



Roughly speaking, detecting the most frequent relations in the domain can at best help in ~25% of questions, and the benefit of relation detection drops off fast with the less frequent relations

Broad-domain relation detection remains a major area of research

Question Decomposition Is Key

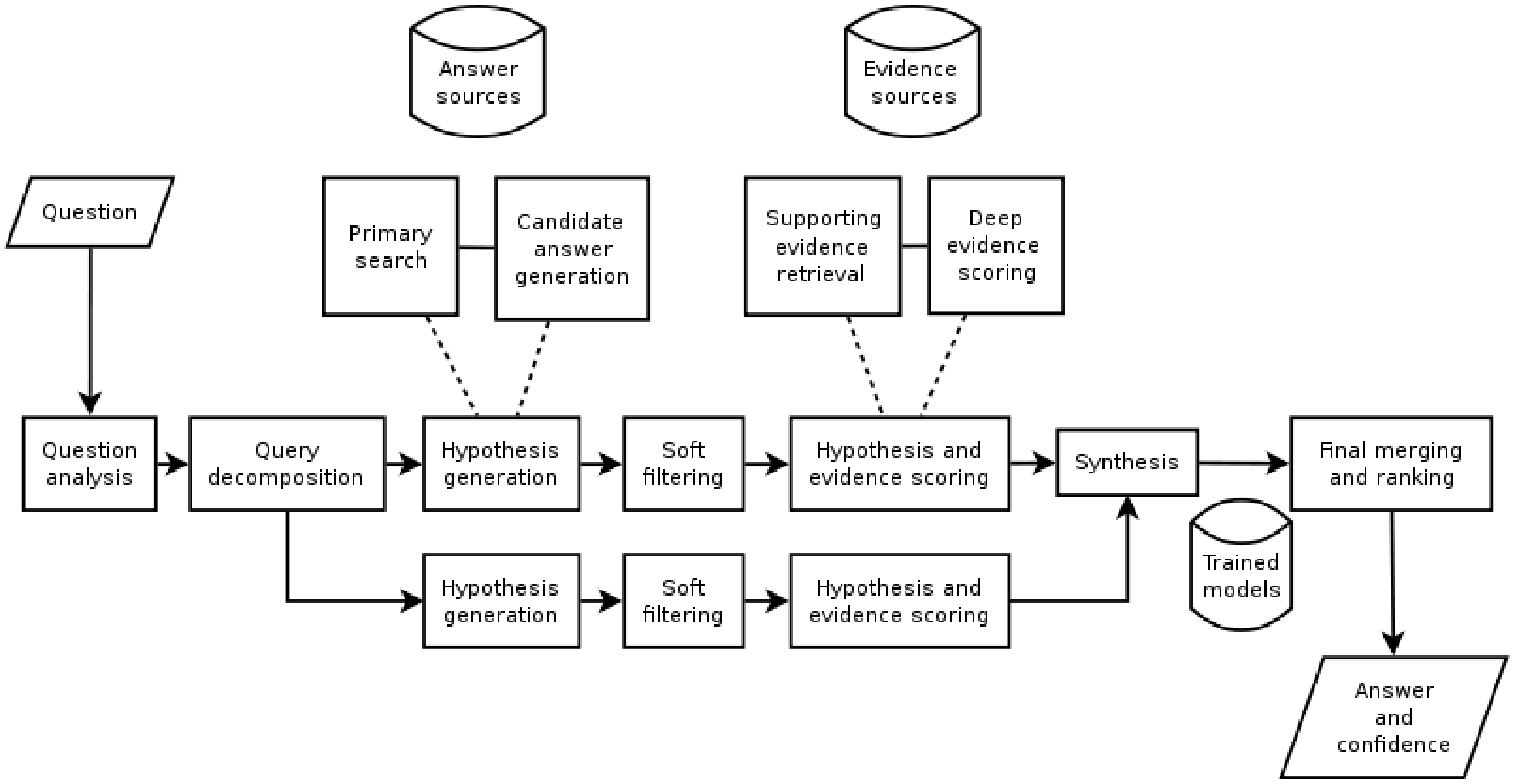
One requirement driven by analysis of Jeopardy clues was the ability to handle questions that are better answered through decomposition

DeepQA uses rule-based deep parsing and statistical classification methods to recognize whether questions should be decomposed & to determine how best to break them up

Embarrassingly Parallel Decomposable Question Life-cycle

We will talk about this pipeline in more detail on the next few slides.

At many points along the path several branches are taken in parallel.



Hypothesis Generation

Hypothesis generation - takes results of question analysis phase & produces candidate answers by searching the system’s sources (we talked about those sources a while ago) & extracting answer-sized snippets from the search results

- Primary Search
- Candidate Answer Generation

Hypothesis Generation 1 of 2: Primary Search

Search performed in hypothesis generation is called “primary search” to distinguish it from search performed during evidence gathering

Goal → Find as many potentially answer-bearing content fragments as possible with the expectation that later phases of content analytics will weed out the right answer from all the candidates

A variety of search techniques used:

- multiple text search engines (ex, Lucene)
- document search
- passage search,
- knowledge base search using SPARQL on triple stores
- generation of multiple search queries for a single question

Hypothesis Generation 2 of 2: Candidate Answer Generation

With copious search results in hand, Watson moves on to candidate generation

Techniques appropriate to the kind of search results we are looking for are applied to generate candidate answers. For instance, Passage search results which require a more detailed analysis of the passage text to identify candidate answers from the passage

If LAT indicated we are looking for names, we seek out candidate answers which are names from the passage. Some sources, such as a triple store and reverse dictionary lookup, produce candidate answers directly

If correct answer(s) are not generated at this stage as candidates, the system has no hope of answering the question → This step significantly favors recall over precision, w/ expectation that later pipeline will tease out a correct answer

System design goal → tolerate noise in early stages of the pipeline and drive up precision downstream. Watson generates several hundred candidate answers at this stage

Soft Filtering: Do not do work that you don't have to.

Key step in managing resource versus precision trade-off is the application of lightweight scoring algorithms to prune the initial, likely large, candidate set before more intensive scoring components are run

Watson combines lightweight analysis scores into a soft filtering score. Candidate answers that pass a soft filter threshold proceed to hypothesis and evidence scoring

The model and threshold are based on machine learning

Watson currently lets ~100 candidates pass the soft filter but this value is tunable

Hypothesis Evidence Retrieval

Each candidate answer that passes the soft filter moves on to the next stage where Watson gathers additional supporting evidence

Watson architecture uses many evidence-gathering techniques

Particularly effective technique for jeopardy is is passage search where the candidate answer is added as a required term to the primary search query derived from the question. This will retrieve passages that contain the candidate answer used in the context of the original question terms.

Hypothesis Scoring

The scoring step is where the bulk of the deep content analysis is performed, and this is where Watson determines the degree of certainty supporting candidate answers

Many different scorers → each consider different dimensions of the evidence

common format for the confidence scores, imposing few restrictions on the semantics of the scores themselves → enables DeepQA devs to rapidly deploy, mix, and tune components to support each other.

Watson employs more than 50 scoring components that produce scores ranging from formal probabilities to counts to categorical features, based on evidence from different types of sources including unstructured text, semistructured text, & triple stores.

Scorers consider things like text passage source reliability, geospatial location, temporal relationships, taxonomic classification, popularity (or obscurity), etc.

Final Ranking and Answer Merging

It is one thing to return documents that contain key words from the question

It is quite another to analyze the question content enough to identify the precise answer

Watson must determine an accurate assessment of confidence

Answer Merging

Without merging, ranking algorithms would be spending time trying to compare multiple generated answers that represent the same concept

To prevent that from happening, Watson uses an ensemble of algorithms to identify equivalent & related hypotheses (ex, Abraham Lincoln and Honest Abe)

Answer Merging

After merging, the system must rank the hypotheses and estimate confidence based on their merged scores

Watson uses a machine-learning approach based on many training questions with known answers

Watson's uses multiple trained models to handle different question classes. Certain scores which may be crucial in identifying the correct answer for a factoid question may not be useful on puzzle questions

Apache UIMA
http://uima.apache.org/

Unstructured
Information Management
Architecture
An Apache Incubator Project

The components in DeepQA are implemented as UIMA annotators

UIMA annotators are software components that analyze text and produce annotations (assertions) about that text

As Watson evolved, it has grown to include hundreds of these components

Example UIMA Annotator Java Implementation

```
public class RoomNumberAnnotator extends JCasAnnotator_ImplBase {
  // create regular expression pattern for Yorktown room number
  private Pattern mYorktownPattern =  Pattern.compile("\\b[0-4]\\d-[0-2]\\d\\d\\b");
  // create regular expression pattern for Hawthorne room number
  private Pattern mHawthornePattern =   Pattern.compile("\\b[G1-4][NS]-[A-Z]\\d\\d\\b");
  public void process(JCas aJCas){
    // The JCas obj is the data obj in UIMA where info is stored. Get doc text from Jcas
    String docText = aJCas.getDocumentText();
    // Search for Yorktown room numbers
    Matcher matcher = mYorktownPattern.matcher(docText);
    int pos = 0;
    while (matcher.find(pos)) {
      // match found – create annotation in the JCas with some additional meta info
      RoomNumber annotation = new RoomNumber(aJCas);
      annotation.setBegin(matcher.start());
      annotation.setEnd(matcher.end());
      annotation.setBuilding("Yorktown");
      annotation.addToIndexes();
      pos = matcher.end();
    }
  }
}
```

Speed and Scale Out: UIMA-AS & OpenJMS

UIMA-AS is part of Apache UIMA

Enables scale out of UIMA applications using asynchronous messaging

Handles all communication, messaging, and queue mgmt necessary using the open JMS standard.

The UIMA-AS deployment of Watson enabled competitive run-time latencies in the 3–5 second range.

http://openjms.sourceforge.net/

Most of the middleware running on Watson, with the notable exception of DB2, is open source.

Additionally we can say that the majority of new software was written in Java and C++

Questions ?

Thank You

IBM *Watson:* *A Hardware & Software Overview*

Eli M. Dow
<emdow@us.ibm.com>