Question Answering Using Semantic Web Technologies

Kwabena Nuamah

Semantic Web Systems

School of Informatics, University of Edinburgh

14 March 2016
Outline

• The question answering problem
• Previous work on QA
• Recent work on QA using Semantic Web approaches
• Challenges with QA
• Rich Inference Framework (RIF)
Question Answering Systems

• Systems that automatically answer questions posed by humans.
• Questions are usually posed in natural language.

• Two forms of question answering:
  • Closed-domain: Domain of questions is specific, e.g. medicine, finance, etc.
  • Open-domain: No restriction on the domain of questions. Questions can be about anything in the world.

• Open-domain question answering requires a lot more general knowledge about the world to accomplish.
The Semantic Web

• It is the focus of this course

• Creation and sharing of ontologies by organizations using Linked Open Data is good.

• LOD makes a lot knowledge available in a semantically-rich format accessible to machines for querying and processing.

• Several government organizations following principles of open data, and some going as far as creating SPARQL endpoints for their data.
  • Example: Statistics Beta by Scottish government (http://statistics.gov.scot/)
Approaches to QA

• Several approaches including:
  • Natural Language Processing (NLP)
  • Logical Reasoning
  • Probabilistic Reasoning
  • Information Retrieval

• Most successful systems have been a hybrid of the techniques above.
• Next, we’ll look at some of the QA systems that have been built (past and present).
QA3 and DEDUCOM

- **QA3** (Green, 1969) is based on theorem proving techniques. Follows QA1 and QA2.
- Example: "Find $x$ such that $P(x)$ is true", where $P$ is a predicate.
- Equivalent to solving $\exists x. P(x)$ in a theorem prover and finding the substitution for $x$.
- Used in *Tower of Hanoi* puzzles and in Robot Problem Solving.

- **DEDUCOM** (Slagle, 1965) (DEDUctive COMmunicator): A deductive QA system created in Lisp.
- System is “told” a set of facts, and it answers questions using those facts.
- Uses a depth-first search procedure for deduction. Process shown in flowchart.
- Very slow.
**START** (Katz et.al, 1988, 2005)

- SynTactic Analysis using Reversible Transformations.
- Uses natural language (NL) annotations to bridge the gap between full text NL QA and sentence-level text analysis.
- START compares user’s query to annotations in the KB.
- **If** match is found between the segment corresponding to the annotations is returned as the answer.

- Uses wide set of KB including the CIA’s *The World Factbook*, and other web knowledge sources.
- Latter revisions use Omnibase, a structured query interface to heterogeneous data on the web.
- Used *object-property-value* model.
AskMSR (Banko et.al, 2002)

- Exploits redundancies in web data by:
  - collecting summaries of the search results,
  - mining and filtering n-grams,
  - determining best answers from remaining data.
OQA (Fader et.al., 2014)

- **Open Question Answering.**
- Factors QA problem into sub-problems including question paraphrasing and query reformulation.
- Maps questions and answers by applying derivation operators: parsing, paraphrase, query-rewrite and execution.
- Uses ten handwritten operators which map question patterns to query patterns.
- Inference task focuses on finding answer with the highest confidence score for all the possible derivations.
Recent QA Systems

- IBM Watson
- Wolfram|Alpha
- Microsoft Cortana
- Google Now
- Apple Siri
- ... etc.
Semantic Web QA Systems

• These systems leverage the semantic web: its formalisms, ontologies and knowledge bases and (or) tools.

• Will discuss some QA systems that have used Semantic Web technologies:
  • ANGIE
  • PowerAqua
  • IBM Watson
  • GORT
  • Rich Inference Framework (RIF)
ANGIE (Preda & Kasneci, 2010)

- Active Knowledge for Interactive Exploration.
- Uses RDF datasets to answer questions.
- ANGIE gathers data from multiple sources to enrich an RDF KB.
- Uses a Query Translation Module that takes a user’s query and translates it into a sequence of function compositions.
- Sends SPARQL queries and web calls to the RDF-3X processor, which combines triples from the local KB and triples from the web.
PowerAqua
(Lopez et.al, 2012)

- Creates query triples from a user query.
- Finds matching triples from its local KB.
- Has a Semantic Storage Platform to connect to different RDF storage systems, e.g. Virtuoso, Sesame, etc.
- Uses a Triple Similarity Service that explores ontological relationships in the KB and searches for the triple that best match the query triple.
- Merges equivalent entities and applies a ranking criteria based on confidence of mapping algorithm.
IBM Watson

- Initially applied to *Jeopardy* quiz game; now being applied to other domains, e.g. medicine, finance, law, etc.
- Uses *DeepQA* [Ferrucci et al, 2010], a pipeline architecture for its QA process.
- Algorithm analyses evidence along different dimensions such as time, geography, popularity, and semantic relatedness.
- Several processes involved: topic analysis, question decomposition, hypothesis generation, hypothesis and evidence scoring, synthesis, confidence merging and ranking, answer and confidence.
- Drew on huge number of disparate approaches from collaborating projects.
- Takes advantage of Semantic Web and Linking Open Data resources (e.g. DBPedia and YAGO) to provide solutions that cover a wide range of domains.
GORT (Bundy et al, 2013)

• Guesstimation with Ontologies and Reasoning Techniques.

• A semi-automatic guesstimation system implemented in SWI-Prolog and Java.

• Solving guesstimation-type questions. E.g.
  “What area of solar panels would be needed to meet the UK's electricity consumption?”

• Searches for facts using SINDICE Semantic Web Search Engine [Tummarello et al, 2007].

• GORT solves problems using a set of proof methods: count, total size, law of averages, distance, rate of change, aggregation over parts, geometry, etc.
Challenges with QA

• Uncertainty from noisy data.
• Difficulty with large knowledge bases from which to find relevant answers.
• Most QA system have largely been focussed on factoid retrieval. Most lack the kind of inference humans make to answer more complex questions.
• Assumptions of pre-stored answers.
• For example, “Was the population of France greater than the population of England in 2007?”
• The factoid that answers this question will very likely not exist in a KB.
• QA systems need to incorporate more kinds of inference mechanisms to tackle these kinds of questions.
Rich Inference Framework

• QA system with “richer” inference mechanisms.

• Focuses on
  • question decomposition strategies,
  • inference methods and
  • answer composition from individual facts.

• Motivated by how to infer novel facts from what we already know.
• Ongoing work by Nuamah, Bundy and Lucas, University of Edinburgh.
Inference Example

- Linear Regression is an example of inference by using existing data to infer (predict) an unknown fact.
Rich Inference

• Reasoning and curation
  • Combine logic-based, graph-based and statistical inference.
  • Exploit semantic web datasets.
  • Normalize data in different formats into the form required by inference strategy.

• Heuristics and commonsense knowledge
  • Background knowledge to guide strategy selection.
  • Commonsense knowledge to augment collected data during inference.

• Uncertainty
  • Deal with noisy and incomplete data.
  • Determine confidence in answer as heuristics and inference strategies are applied to facts.
  • Convey uncertainty to user in an intelligible way.
QA Pipeline

1. User Query
2. Query analysis and Feature Extraction
3. Formal Representation
4. Decomposition strategy selection
5. Decomposition tree exploration & search query generation
6. Knowledge (Fact) Search
7. Local Curation of Search results (alignment and merging)
8. Inference
9. Answer Synthesis

Rich Inference

[Unsuccessful]

[Successful]
Inference Model in QA

Human expert defines or chooses the “program” to answer the question T.

Human expert selects the knowledge base

Typical QA Approach

- Emphasis on the design and optimization of \( \Sigma \) to get the best possible answer from available data.
- Question is impossible to answer if the particular program selected does not fit the question or the data.
Our Model using RIF

- Reason over available inference methods as well as data to answer a question.
- Integrate both programs and data in the inference process.

Each $\Sigma_i$ represents an alternative strategy to decompose the question by some dimension such as time, place, etc. based on feature in question.
RIF Representation

• RIF is graph-based.
• 3 types of vertices:
  • Tasks (Queries)
  • Programs (Decomposition Strategies and Inference programs)
  • Facts (Data)

• Decomposition strategies include:
  • Temporal (using regression)
  • Geo-spatial
  • Ratios
  • Rate of change
RIF Decomposition
"Which country has the largest female population in Europe."

\[
\text{argmax}\{x | \text{country}(x) \land \text{loc}(x, \text{Europe})\}, \lambda x. (\text{female\_population}(x, y) \land \text{instance\_date}(y, 2025))
\]

Each node is decomposed further.
Current Implementation

• Built in Java
• Off-the-shelf libraries/components include:
  • Apache Jena (https://jena.apache.org/)
  • Fuseki (https://jena.apache.org/documentation/fuseki2/)
  • WordNet (https://wordnet.princeton.edu/)
  • ConceptNet (http://conceptnet5.media.mit.edu/)
  • Spark (http://sparkjava.com/)
  • Apache Commons Math (http://commons.apache.org/proper/commons-math/)
• Launched either as a command-line application or a web service.
Conclusion

• Rich Inference Framework (RIF) integrates
  • decomposition strategies,
  • inference programs and
  • facts
  in the reasoning process.
• RIF is graph-based and allows concurrent search for answers using different strategies.
• RIF decompositions can be query-driven or fact-driven.

• RIF goes beyond simple factoid retrieval, to use recursive decomposition of queries and application of statistical inference methods to infer novel facts, then propagate them up the graph.
• Extends the range of question that QA systems can handle.


