(TREC-style)
Question Answering systems

Christopher Manning
CS224N/Ling 237 2004

(Revised title borrowed from Sanda Harabagiu, UMass
Nicholas Kushmerick)

Question Answering from text

- An idea originating from the IR community
- With massive collections of full-text documents, simply finding relevant documents is of limited use: we want answers from texthashes
- QA: give the user a (short) answer to their question, perhaps supported by evidence.
- The common person’s view? [From a novel]
  - “Like the internet, really I do. Anytime I need a piece of..."[from a novel]
- “I’m the first guy to get the modern humming, but as a source of information, it..."[from a novel]

People want to ask questions...

Examples from AltaVista query log

- who invented surf music?
- how to make stink bombs
- where are the snowduns of yesteryear?
- which english translation of the bible is used in official catholic liturgies?
- how to do clayart
- how to copy psx
- how tall is the sears tower?
- where can I find information on puritan religion?
- what are the 7 wonders of the world
- how can I eliminate stress
- What vacuum cleaner does Consumers Guide recommend
- Around 1G-15% of query logs

AskJeeves

- AskJeeves is probably most hyped example of "Question answering"
- It largely does pattern matching to match your question to their own knowledge base of questions
- If that works, you get the human-curated answers to that known question
- If that fails, it falls back to regular web search
- A potentially interested middle ground, but a fairly weak shadow of real QA

A Brief (Academic) History

- In some sense question answering is not a new research area
- Question answering systems can be found in many areas of NLP research, including:
  - Natural language database systems
  - A lot of early NLP work on these
  - Spoken dialog systems
  - Currently very active and commercially relevant
- The focus on open-domain QA is new
  - MURAX (Kupiec 1993): Encyclopedia answers
  - Hirschman: Reading comprehension tests
  - TREC QA competition: 1999-

Online QA System Examples

- Examples
  - AnswerBus is an open-domain question answering system: www.answerbus.com
  - Ionaut: http://www.ionaut.com:8400/
  - LCC: http://www.languagecomputer.com/
  - EasyAsk, AnswerLogic, AnswerFriend, Start, Quasm, Mulder, Webclopedia, etc.
  - ISI TextMap
    http://brahms.isi.edu:8080/textmap/
1. Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?
2. What was the monetary value of the Nobel Peace Prize in 1989?
3. What does the Peugeot company manufacture?
4. How much did Mercury spend on advertising in 1993?
5. What is the name of the managing director of Apricot Computer?
6. Why did David Koresh ask the FBI for a word processor?
7. What debts did Qintex group leave?
8. What is the name of the rare neurological disease with symptoms such as: involuntary movements (tics), swearing, and incoherent vocalizations (grunts, shouts, etc.)?
Top Performing Systems

- Currently the best performing systems at TREC can answer approximately 70% of the questions.
- Approaches and successes have varied a fair deal.
- Knowledge-rich approaches, using a vast array of NLP techniques, have been successful.
  - Notably Herbagiu, Moldovan et al. - SMU/UF/D/LSECC.
  - AskMSR system stressed how much could be achieved by very simple methods with enough text (and now various copies).
- Middle ground is to use a large collection of surface matching patterns (ISI).

AskMSR

- Web Question Answering: Is More Always Better?
  - Dumais, Banko, Brill, Lin, Ng (Microsoft, MIT, Berkeley)
- Q: "Where is the Louvre located?"
  - Want "Paris" or "France" or "75058 Paris Cedex 01" or a map
  - Don't just want URLs

AskMSR: Shallow approach

- In what year did Abraham Lincoln die?
- Ignore hard documents and find easy ones

AskMSR: Details

Step 1: Rewrite queries

- Intuition: The user's question is often syntactically quite close to sentences that contain the answer.
  - Where is the Louvre Museum located?
  - The Louvre Museum is located in Paris
  - Who created the character of Scrooge?
  - Charles Dickens created the character of Scrooge.

Query rewriting

- Classify question into seven categories
  - What is/are/was/are/were...?
  - Where is/are/was/were...
  - When is/are/was/were...
  - Which is/are/was/were...
  - How is/are/was/were...
  - For what questions, move to all possible locations?
    - "Where is the Louvre Museum located?" → "The Louvre Museum located" → "the Louvre Museum located" → "the Louvre Museum is located" → "the Louvre Museum located is" → "the Louvre Museum located in"
  - Expected answer "Datatype" (eg. Date, Person, Location, Time)
    - What was the French Revolution? → DATE
- Hand-crafted classification/rewrite/datatype rules
(Could they be automatically learned?)
**Query Rewriting – weights**

- One wrinkle: Some query rewrites are more reliable than others
- Where is the Louvre Museum located?
- Weight 1: Lots of non-answers could come back too
- Weight 5: if we get a match, it’s probably right
- “the Louvre Museum is located”

**Step 2: Query search engine**

- Send all rewrites to a Web search engine
- Retrieve top N answers (100?)
- For speed, rely just on search engine’s “snippets”, not the full text of the actual document

**Step 3: Mining N-Grams**

- Unigram, bigram, trigram, ... N-gram: list of N adjacent terms in a sequence
  - Unigrams: Web, Question, Answering, Is, More, Always, Better

**Mining N-Grams**

- Simple: Enumerate all N-grams (N=1,2,3) in all retrieved snippets
  - Use hash table and other fancy footprint to make this efficient
  - Weight of an n-gram: occurrence count, each weighted by “reliability” (weight of rewrite that fetched the document)
- Example: “Who created the character of Scrooge?”
  - Dickens = 117
  - Christmas Carol = 78
  - Charles Dickens = 75
  - Disney = 72
  - Carol Banks = 54
  - A Christmas = 41
  - Christmas Carol = 45
  - Uncle = 31

**Step 4: Filtering N-Grams**

- Each question type is associated with one or more “data-type filters” = regular expression
- When...
- Where...
- What...
- Location
- Who...
- Person
- Boost score of n-grams that do match regexp
- Lower score of n-grams that don’t match regexp
- Details omitted from paper....

**Step 5: Tiling the Answers**

<table>
<thead>
<tr>
<th>Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Charles Dickens
Charles Dickens
Mr Charles

- N-Grams: merged, discard old n-grams
- Tiling highest-scoring n-gram
- Repeat, until no more overlap

---

**Mr Charles Dickens**
Results

- Standard TREC contest test-bed: ~1M documents; 900 questions
- Technique doesn’t do too well (though would have placed in top 9 of ~30 participants!)
  - MRR = 0.262 (i.e., right answer ranked about #4–#5 on average)
  - Why? Because it relies on the enormity of the Web
  - Using the Web as a whole, not just TREC’s 1M documents... MRR = 0.42 (i.e., on average, right answer is ranked about #2–#3)

Issues

- In many scenarios (e.g., monitoring an individuals email...) we only have a small set of documents
- Works best only for “Trivial Pursuit”-style fact-based questions
- Limited/brittle repertoire of question categories
  - answer data types/filters
  - query rewriting rules

ISI (Ravichandran and Hovy 2002): Surface patterns approach

- Use of Characteristic Phrases
  - “When was <person> born”
    - Typical answers
      - “Mozart was born in 1756.”
      - “Gandhi (1869–1948)...”
    - Suggests phrases like
      - “<NAME> was born in <BIRTHDATE>”
      - “<NAME> ( <BIRTHDATE> )”
    - as Regular Expressions can help locate correct answer

Use Pattern Learning

- Example:
  - “The great composer Mozart (1756–1791) achieved fame at a young age”
  - “Mozart (1756–1791) was a genius”
- “The whole world would always be indebted to the great music of Mozart (1756–1791)”
  - Longest matching substring for all 3 sentences is “Mozart (1756–1791)”
  - Suffix tree would extract “Mozart (1756–1791)” as an output, with score of 3
  - Reminiscent of IE pattern learning

Pattern Learning (cont.)

- Repeat with different examples of same question type
  - “Gandhi 1869”, “Newton 1642”, etc.
- Some patterns learned for BIRTHDATE
  - a. born in <ANSWER>, <NAME>
  - b. <NAME> was born on <ANSWER>,
  - c. <NAME> ( <ANSWER> )
  - d. <NAME> ( <ANSWER> – )

Experiments (Ravichandran and Hovy 2002)

- 6 different Question types
  - from Webclopedia QA Typology (Hovy et al., 2002a)
    - BIRTHDATE
    - LOCATION
    - INVENTOR
    - DISCOVERER
    - DEFINITION
    - WHY-FAMOUS
Experiments: pattern precision

- BIRTHDATE table:
  - 1.0 <NAME> (<ANSWER> - )
  - 0.85 <NAME> was born on <ANSWER>,
  - 0.6 <NAME> was born in <ANSWER>,
  - 0.59 <NAME> was born <ANSWER>,
  - 0.53 <ANSWER> <NAME> was born
  - 0.50 - <NAME> (<ANSWER> - ),
  - 0.36 <NAME> (<ANSWER> - )
- INVENTOR
  - 1.0 <ANSWER> invents <NAME>
  - 1.0 the <NAME> was invented by <ANSWER>,
  - 1.0 <ANSWER> invented the <NAME> in

Experiments (cont.)

- WHY-FAMOUS
  - 1.0 <ANSWER> <NAME> called
  - 1.0 locate <ANSWER> <NAME> of
  - 0.71 <NAME> is the <ANSWER> of
- LOCATION
  - 1.0 <ANSWER>’s <NAME>
  - 1.0 regional: <ANSWER> : <NAME>
  - 0.92 near <NAME> in <ANSWER>
- Depending on question type, get high MRR (0.6-0.9), with higher results from use of
  Web than TREC QA collection

Shortcomings & Extensions

- Need for POS &/or semantic types
  - "Where are the Rocky Mountains?"
  - "Denver’s new airport, topped with white fiberglass
cores in imitation of the Rocky Mountains in the
background, located 10 miles east of the city of Denver"
  - <NAME> in <ANSWER>
- NE tagger &/or ontology could enable
  system to determine "background" is not a
  location

Shortcomings... (cont.)

- Long distance dependencies
  - "Where is London?"
  - "London, which has one of the most bustling
airports in the world, lies on the banks of the river
The Thames"
  - would require pattern like: <ANSWER>, <<vary_word>> lie on <NAME>
- Abundance & variety of Web data helps
  system to find an instance of patterns w/o
  losing answers to long distance dependencies

Shortcomings... (cont.)

- System currently has only one anchor word
- Doesn’t work for Q types requiring multiple
  words from question to be in answer
  - "In which county does the city of Long Beach lie?"
  - "Long Beach is situated in Los Angeles County"
  - required pattern:
    - <Q_TERM_1> is situated in <ANSWER> <Q_TERM_2>
- Does not use case
  - "What is a micron?"
  - "a spokesperson for Micron, a maker of
seminconductors, said SIMMs are...
- If Micron had been capitalized in question,
  would be a perfect answer

LCC: Harabagiu, Moldovan et al.
Value from sophisticated NLP - Pasca and Harabagiu (2001)

- Good IR is needed; SMART paragraph retrieval
- Large taxonomy of question types and expected answer types is crucial
- Statistical parser used to parse questions and relevant text for answers, and to build KB
- Query expansion loops (morphological, lexical synonyms, and semantic relations) important
- Answer ranking by simple ML method

Answer types in State-of-the-art QA systems

QA Typology (from ISI USC)

- Typology of typical Q forms—94 node (47 leaf nodes)
- Analyzed 17,384 questions (from answers.com)

Extracting Answers for Factoid Questions

- In TREC 2003 the LCC QA system extracted 289 correct answers for factoid questions
- The Name Entity Recognizer was responsible for 234 of them

Special Case of Names

Questions asking for names of authored works

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1934: What is the play 'West Side Story' based on?</td>
<td>Answer: Three Fugues and a Study</td>
</tr>
<tr>
<td>2934: What is the motto for the Boy Scouts?</td>
<td>Answer: Be Prepared</td>
</tr>
<tr>
<td>3036: What is the notice for the Leo Strauss?</td>
<td>Answer: Teaching Music Today</td>
</tr>
<tr>
<td>Dog: What year finally ended WRF?</td>
<td>Answer: Versatilis</td>
</tr>
<tr>
<td>2338: What American landmark stands on Liberty Island?</td>
<td>Answer: Statue of Liberty</td>
</tr>
</tbody>
</table>

NE-driven QA

- The results of the past 5 TREC evaluations of QA systems indicate that current state-of-the-art QA is determined by the recognition of Named Entities:
  - Precision of recognition
  - Coverage of name classes
  - Mapping into concept hierarchies
  - Participation into semantic relations (e.g., predicate-argument structures or frame semantics)
Concept Taxonomies

- For 29% of questions the QA system relied on an offline taxonomy with semantic classes such as:
  - Disease
  - Drugs
  - Colors
  - Insects
  - Games
- The majority of these semantic classes are also associated with patterns that enable their identification

Abductive inference

- System attempts inference to justify an answer (often following lexical chains)
- Their inference is a kind of funny middle ground between logic and pattern matching
- But quite effective: 30% improvement
- Q: When was the internal combustion engine invented?
- A: The first internal combustion engine was built in 1867.

Question Answering Example

- How hot does the inside of an active volcano get?
- get(TEMPERATURE, inside(volcano(active)))
- "lava fragments belched out of the mountain were as hot as 300 degrees Fahrenheit"
- fragments(lava, TEMPERATURE(degrees(300)), belched(out, mountain))
- volcano BA mountain
- lava MPARTOF volcano
- lava inside volcano
- The needed semantic information is in WordNet definitions, and was successfully translated into a form that was used for rough 'proofs'

The Architecture of LCC's QA System around 2003

Definition Questions

- They asked about:
  - PEOPLE (most of them starting with 'Who')
  - other types of NAMES
  - general concepts
- People questions
  - Many use the PERSON name in the format [First name, Last name]
  - sometimes in the format [First name, Last name, First name]
  - Some names had the PERSON name in format [First name, Last name, Last name]
  - Example: Atlantic City casino
  - Other names had the name as a single word, e.g. very well known person
    - Example: John or Jane Doe
  - Some questions refer to names of kings or princesses:
    - Example: The Queen of England, aka the Queen of Great Britain
**Answering definition questions**

- Most definitions are between OOV patients
- The new population

<table>
<thead>
<tr>
<th>Question</th>
<th>Definition</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Complex questions**

- Characterized by the need of domain knowledge
- There is no single answer type that can be identified, but rather an answer-structure needs to be recognized
- Answer selection becomes more complex, since inference based on the semantics of the answer type needs to be evaluated
- Complex questions need to be decomposed into a set of simpler questions

**Example of Complex Question**

How have thefts impacted on the safety of Russia's nuclear navy, and has the theft problem been increased or reduced over time?

Need of domain knowledge

- Definition questions:
- What is theft or nuclear navy?
- What does 'theft or nuclear navy' mean?
- How does one define the increase or decrease of a problem?
- Typical question:
- What is the magnitude of thefts that are likely to be reported?
- What sort of thefts have been reported?
- Navigating questions:
- What is theft or nuclear navy? Only statues, or also former Soviet facilities or, non-Russian equivalents?

**Semantic inference for Q/A**

- The problem of classifying questions
  - E.g., "manner question":
  - Example: "How did Hitler die?"
- The problem of recognizing answer types/structures
  - Should "manner of death" be considered an answer type?
  - What other manner of evaluation should be considered as an answer type?
- The problem of extracting/justifying/generating answers to complex questions
  - Should we learn to extract "manner relations?"
  - What other type of relations should we consider?
  - Is relation recognition sufficient for answering complex questions? Is it necessary?

**Manner-of-death**

In previous TREC evaluations 31 questions asked about manner of death:

- "How did Hitler die?"
- "What is the definition of cancer?"
- "What is the cause of a disease?"
- "What is the main reason for death?"
- "What is the cause of death?"

**Practical Hurdle**

- Not all MANNER-OF-DEATH concepts are identifiable as a verb
  - In cases possessing adjectives that represent the phrases such cases
- Need: a set of patterns
  - Differentiates corresponding to each pattern
  - Well-known E technique (ECAV, Ribuffe)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>X [DIE (be killed)]</td>
<td>in ACCIDENT (ACCIDENT) car crash</td>
</tr>
<tr>
<td>X [DIE (from)] DISEASE</td>
<td>(DISEASE) cancer</td>
</tr>
<tr>
<td>X [DIE (after suffering)] CONDITION</td>
<td>(CONDITION) stroke</td>
</tr>
</tbody>
</table>

- Results: 100 patterns were discovered
Applying Frame Structures to QA

**Parsing Questions**
- What kind of materials were stolen from the Russian navy?

**Target-Predicate (P1): theft**; in 1/96, approximately 7 kg of HEU was reportedly stolen from a naval base in Sovereigns Great.

**Target-Predicate (P2): stolen**; in 1/96, approximately 7 kg of HEU was reportedly stolen from a naval base in Sovereigns Great.

**FS:** What [GOODS (P1): nuclear] [VICTIM (P1): Russia's Pacific Fleet] has also fallen prey to [Target-Predicate (P1): theft]; in 1/96, approximately 7 kg of [GOODS (P2): kind of nuclear materials] [VICTIM (P2): from the Russian Navy] [SOURCE (P2): in Soketskaya Gavan] was reportedly [Target-Predicate (P2): stolen].

Additional types of relations

- **Temporal relations**
- **Causal relations**
- **Evidential relations**
- **Part-whole relations**

... plenty of research to be done to be able to answer these kinds of questions!

References

- AskMSR: Question Answering Using the Worldwide Web
  - Michele Banko, Eric Brill, Susan Dumais, Jimmy Lin
- Web Question Answering: Is More Always Better?
  - Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, Andrew Ny

References