Question Answering

- Following largely from Chris Manning’s slides, which includes slides originally borrowed from Sanda Harabagiu, ISI, Nicholas Kushmerick.
Question Answering from Text

- **Question Answering:**
  - Give the user a (short) answer to their question, perhaps supported by evidence.
  - 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 textbases

- **The common person’s view? [From a novel]**
  - “I like the Internet. Really, I do. Any time I need a piece of shareware or I want to find out the weather in Bogota … I’m the first guy to get the modem humming. But as a source of information, it sucks. You got a billion pieces of data, struggling to be heard and seen and downloaded, and anything I want to know seems to get trampled underfoot in the crowd.”

People *want* to ask questions?

**Examples from AltaVista query log**
- who invented surf music?
- how to make stink bombs
- where are the snowdens 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?

**Examples from Excite query log (12/1999)**
- how can i find someone in texas
- 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 10–15% of query logs**
AskJeeves (Classic)

- Probably the 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 interesting middle ground, but not full QA

A Brief (Academic) History

- 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–
Question Answering at TREC

- Question answering competition at TREC consists of answering a set of 500 fact-based questions, e.g., “When was Mozart born?”.  
- For the first three years systems were allowed to return 5 ranked answer snippets (50/250 bytes) to each question.  
  - IR think  
  - Mean Reciprocal Rank (MRR) scoring:  
    - 1, 0.5, 0.33, 0.25, 0.2, 0 for 1, 2, 3, 4, 5, 6+ doc  
  - Mainly Named Entity answers (person, place, date, …)  
- From 2002 the systems are only allowed to return a single exact answer and the notion of confidence has been introduced.

The TREC Document Collection

- One recent round: news articles from:  
  - AP newswire, 1998-2000  
  - Xinhua News Agency newswire, 1996-2000  
- In total 1,033,461 documents in the collection.  
- 3GB of text  
- While small in some sense, still too much text to process entirely using advanced NLP techniques  
- Systems usually have initial information retrieval followed by advanced processing.  
- Many supplement this text with use of the web, and other knowledge bases
Sample TREC questions

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 stole the show in 2000, 2001
    - Notably Harabagiu, Moldovan et al. – SMU/UTD/LCC
  - AskMSR system stressed how much could be achieved by very simple methods with enough text (and now various copycats)
  - Middle ground is to use large collection of surface matching patterns (ISI)
Online QA System 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/

---

Webclopedia Architecture

IR
- Steps: center query from question (WordNet-expanded) or reuse top 1000 documents
- Engine: MG (Sydney)—(Lin)
  AT&T (TREC)—(Lin)

Segmentation
- Steps: segment each document into topical segments
- Engine: fixed-length (not used)
  TextMap (Fawcett 00)—(Lin)
  C99 (Chin 99)—(Lin)
  MAXNET (Lin 00; not used)

Ranking
- Steps: score each sentence in each segment, using WordNet expansion
  Engine: Fastfinder (Lin)

Matching
- Steps: match general concept patterns against parse trees
  match desired semantic type against parse tree elements
  match desired words against words in sentences
  Engine: matcher (Rank)

Ranking and answer extraction
- Steps: rank candidate answers; extract and format them
  Engine: part of matcher (Rank)

Question parsing
- Steps: parse question, find desired semantic type
  Engine: Identifiers (Lin)
  CONTEX (Stamatatos)

Semantics Parsing
- Steps: parse segment sentences
  Engine: CONTEX (Stamatatos)

QA typology
- Categorize QA types in taxonomy (Gardner)

Constraint patterns
- Identify likely answers in relation to other parts of the sentence (Gardner)
At the time, these were the best 10 answers found for the question:

1. 556 4239 When John Lennon died in 1980, the midlife childhood experience which took place 13 years earlier... (http://www.storms.com/whale.htm)
2. 556 4239 When John Lennon died in 1980, the midlife childhood experience which took place 13 years earlier... (http://www.storms.com/whale.htm)
3. 556 1782 John Lennon died in 1980, which is both of that era... (http://www.google.com/search?q=John+Lennon+1980)
4. 556 1782 As many people know, John Lennon died in 1980 and... (http://www.google.com/search?q=John+Lennon+1980)
5. 556 1782 As many people know, John Lennon died in 1980 and... (http://www.google.com/search?q=John+Lennon+1980)
6. 556 1782 John Lennon died in 1980 when he was shot by a fan outside his New York City... (http://www.google.com/search?q=John+Lennon+1980)
7. 556 1782 John Lennon died in 1980 when he was shot by a fan outside his New York City... (http://www.google.com/search?q=John+Lennon+1980)
8. 556 1782 John Lennon died in 1980 when he was shot by a fan outside his New York City... (http://www.google.com/search?q=John+Lennon+1980)
9. 556 1782 John Lennon died in 1980 when he was shot by a fan outside his New York City... (http://www.google.com/search?q=John+Lennon+1980)
10. 556 1782 John Lennon died in 1980 when he was shot by a fan outside his New York City... (http://www.google.com/search?q=John+Lennon+1980)

Current top 10 (of 109) for "Who was the prime minister of Australia in 1990?"... still finding more...

2. 203 09271 sound for Prime Minister Bill Hunter special speech, Australia Today - 1990... (http://www.mca.gov.au/faces/australia)
3. 203 09271 John Howard at the prime minister of Australia, Nickie on something he's essentially stopped in, though he managed to eng... (http://www.dpi.nsw.gov.au/dpi/campaigns/australia/campaign.html)
5. 203 09271 Visit to Japan by Prime Minister John Howard December 2003... (http://www.mca.gov.au/govt/ministers/australia)
6. 203 09271 Visit to Japan by Prime Minister John Howard December 2003... (http://www.mca.gov.au/govt/ministers/australia)
7. 203 09271 Mr. Howard the prime minister of Australia, the Prime Minister's Official Residence... (http://www.mca.gov.au/govt/ministers/australia)
8. 203 09271 Visit to Japan by Prime Minister John Howard December 2003... (http://www.mca.gov.au/govt/ministers/australia)
10. 203 09271 Visit to Japan by Prime Minister John Howard December 2003... (http://www.mca.gov.au/govt/ministers/australia)
The Google Answer #1

- Include question words etc. in your stop-list
- Do standard IR

Sometimes this (sort of) works:

**Question:** Who was the prime minister of Australia during the Great Depression?

**Answer:** James Scullin (Labor) 1929–31.
But often it doesn’t…

- **Question:** How much money did IBM spend on advertising in 2002?
- **Answer:** I dunno, but I’d like to …
The Google Answer #2

- Take the question and try to find it as a string on the web
- Return the next sentence on that web page as the answer
- Works brilliantly if this exact question appears as a FAQ question, etc.
- Works lously most of the time
- Reminiscent of the line about monkeys and typewriters producing Shakespeare
- But a slightly more sophisticated version of this approach has been revived in recent years with considerable success...
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
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: Variations

- Classify question into seven categories
  - Who is/was/are/were...?
  - When is/did/will/are/were ...?
  - Where is/are/were ...
  
  a. Category-specific transformation rules
     eg “For Where questions, move ‘is’ to all possible locations”
     “Where is the Louvre Museum located”
     → “is the Louvre Museum located”
     → “the is Louvre Museum located”
     → “the Louvre is Museum located”
     → “the Louvre Museum is located”
     → “the Louvre Museum located is”
  
  b. Expected answer “Datatype” (eg, Date, Person, Location, …)
     When 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”

+Louvre +Museum +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

- Simple: Enumerate all N-grams (N=1,2,3 say) in all retrieved snippets
- 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
  - Carl 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 …
  - Who …

- 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

Scores

<table>
<thead>
<tr>
<th>Scores</th>
<th>N-Grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Charles Dickens</td>
</tr>
<tr>
<td>15</td>
<td>Dickens</td>
</tr>
<tr>
<td>10</td>
<td>Mr Charles</td>
</tr>
</tbody>
</table>

merged, discard old n-grams

Score 45 | Mr Charles Dickens

N-Grams tile highest-scoring n-gram
N-Grams Repeat, until no more overlap
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 (ie, right answered 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 (ie, on average, right answer is ranked about #2-#3)

Issues

- In many scenarios (e.g., monitoring an individual’s 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
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

Example: Start with “Mozart 1756”
- Results:
  - “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: (R+H, 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 laureate `<ANSWER>` `<NAME>`
  - 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 cones in imitation of the Rocky Mountains in the background, continues to lie empty"
  - <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 busiest airports in the world, lies on the banks of the river Thames"
  - would require pattern like: <QUESTION>, (<any_word>)*, lies on <ANSWER>

- But: abundance & variety of Web data helps system to find an instance of patterns w/o losing answers to long distance dependencies
Shortcomings… (cont.)

- Their system uses 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 \(<\text{ANSWER}>\ <Q\_TERM\_2>\)

- Does not use case
  - “What is a micron?”
  - “…a spokesman for Micron, a maker of semiconductors, said SIMMs are…”

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 SOA QA systems

- Labels questions with answer type based on a taxonomy
- Classifies questions (e.g. by using a maximum entropy model)
QA Typology (from ISI USC)

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

Extracting Answers for Factoids

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

<table>
<thead>
<tr>
<th>Year</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1934</td>
<td>What is the play “West Side Story” based on?</td>
<td>Romeo and Juliet</td>
</tr>
<tr>
<td>1976</td>
<td>What is the motto for the Boy Scouts?</td>
<td>Be Prepared</td>
</tr>
<tr>
<td>1982</td>
<td>What movie won the Academy Award for best picture in 1989?</td>
<td>Driving Miss Daisy</td>
</tr>
<tr>
<td>2080</td>
<td>What peace treaty ended WWI?</td>
<td>Versailles</td>
</tr>
<tr>
<td>2102</td>
<td>What American landmark stands on Liberty Island?</td>
<td>Statue of Liberty</td>
</tr>
</tbody>
</table>

Named Entity Recognition for 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)
For 29% of questions the QA system relied on an off-line 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.

Syntax to Logical Forms

- Syntactic analysis plus semantic => logical form
- Mapping of question and potential answer LFs to find the best match
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.
- invent -> create Mentally -> create -> build

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 ISA mountain
  - lava ISPARTOF volcano ■ lava inside volcano
  - fragments of lava HAVEPROPERTIESOF lava
- The needed semantic information is in WordNet definitions, and was successfully translated into a form that was used for rough ‘proofs’
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]
    - examples: Aaron Copland, Allen Iverson, Albert Ghiorso  
  - Some names had the PERSON name in format [First name, Last name1, Last name2]
    - example: Antonia Coello Novello  
  - Other names had the name as a **single word** → very well known person
    - examples: Nostradamus, Absalom, Abraham  
  - Some questions referred to names of kings or princes:
    - examples: Vlad the Impaler, Akbar the Great
Answering definition questions

- Most QA systems use between 30-60 patterns
- The most popular patterns:

<table>
<thead>
<tr>
<th>Id</th>
<th>Pattern</th>
<th>Freq. Usage</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>person-hyponym QP</td>
<td>0.43%</td>
<td>The doctors also consult with former Italian Olympic skier Alberto Tomba, along with other Italian athletes</td>
</tr>
<tr>
<td>9</td>
<td>QP, the AP</td>
<td>0.28%</td>
<td>Bausch Lomb, the company that sells contact lenses, among hundreds of other optical products, has come up with a new twist on the computer screen magnifier</td>
</tr>
<tr>
<td>11</td>
<td>QP, a AP</td>
<td>0.11%</td>
<td>ETA, a Basque language acronym for Basque Homeland and Freedom, has killed nearly 800 people since taking up arms in 1968</td>
</tr>
<tr>
<td>13</td>
<td>QA, an AP</td>
<td>0.02%</td>
<td>The kidnappers claimed they are members of the Abu Sayaf, an extremist Muslim group, but a leader of the group denied that</td>
</tr>
<tr>
<td>21</td>
<td>AP such as QP</td>
<td>0.02%</td>
<td>For the hundreds of Albanian refugees undergoing medical tests and treatments at Fort Dix, the news is mostly good: Most are in reasonable health, with little evidence of infectious diseases such as TB</td>
</tr>
</tbody>
</table>

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**

To what degree do different thefts put nuclear or radioactive materials at risk?

**Question decomposition**

**Definition questions:**
- What is meant by nuclear navy?
- What does ‘impact’ mean?
- How does one define the increase or decrease of a problem?

**Factoid questions:**
- What is the number of thefts that are likely to be reported?
- What sort of items have been stolen?

**Alternative questions:**
- What is meant by Russia? Only Russia, or also former Soviet facilities in non-Russian republics?
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 complicated, since inference based on the semantics of the answer type needs to be activated
- Complex questions need to be decomposed into a set of simpler questions