Statistical NLP
Spring 2010

Lecture 24: Question Answering

Dan Klein – UC Berkeley

Question Answering

- Following largely from Chris Manning's slides, which includes slides originally borrowed from Sanda Harabagiu, ISI, Nicholas Kushmerick.

Question Answering

- Question Answering:
  - More than search
  - Ask general comprehension questions of a document collection
  - Can be really easy: "What's the capital of Wyoming?"
  - Can be harder: "How many US states' capitals are also their largest cities?"
  - Can be open ended: "What are the main issues in the global warming debate?"
- SOTA: Can do factoids, even when text isn't a perfect match

People *want* to ask questions?

Examples of search queries

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?
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 did 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 (which are presumably good)
- 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
  - New York Times 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 using advanced NLP techniques (on the fly at least)
- Systems usually have initial information retrieval followed by advanced processing.
- Many supplement this text with use of the web, and other knowledge bases

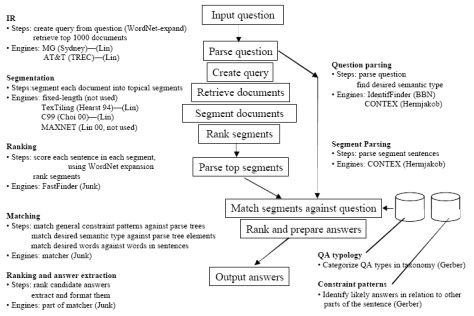
Sample TREC questions

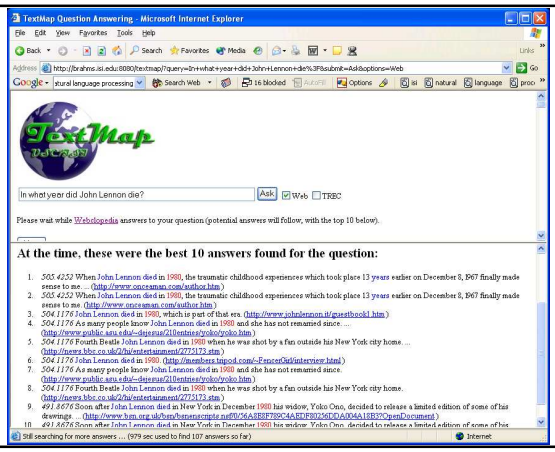
1. Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?
2. What was the monetary value of the Nobel Peace Prize in 1989?
3. What does the Peugeot company manufacture?
4. How much did Mercury spend on advertising in 1993?
5. What is the name of the managing director of Apricot 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.)?

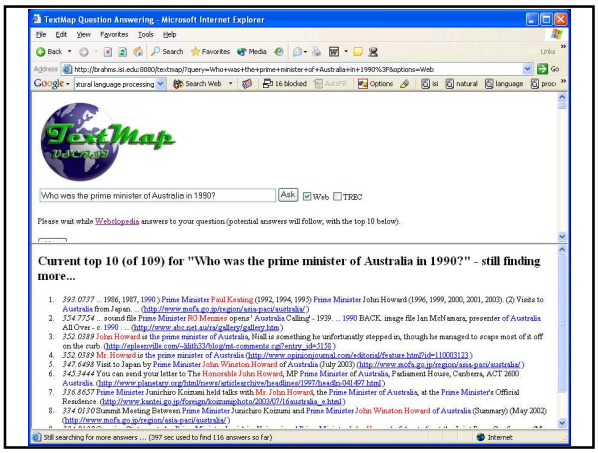
Top Performing Systems

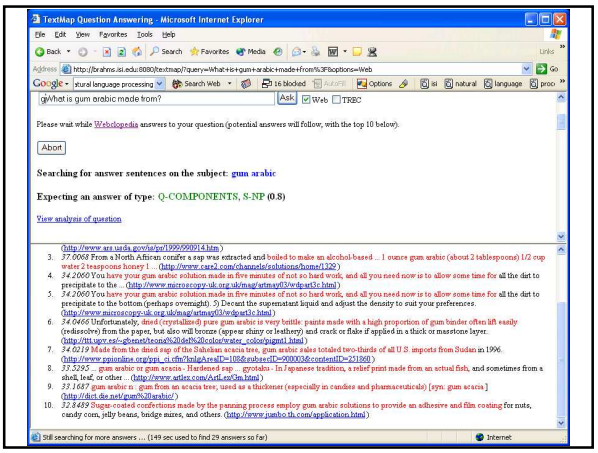
- Currently the best performing systems at TREC can answer approximately 70% of the questions
- Approaches and successes have varied a fair deal
  - Knowledge-rich approaches, using a vast array of NLP techniques stole the show in 2000, 2001, still do well
    - Notably Harabagiu, Moldovan et al. – SMU/UTD/LCC
  - 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)

Webclopedia Architecture









Ravichandran and Hovy 2002 Learning Surface Patterns

- 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: 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.)

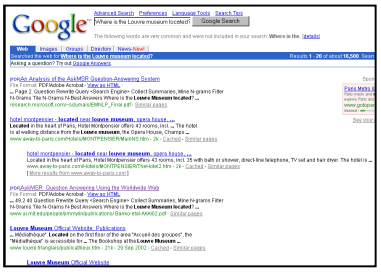
- Long distance dependencies
  - "Where is London?"
  - "London, which has one of the busiest airports in the world, lies on the banks of the river Thames"
  - would require pattern like:
    <QUESTION>, (<any_word>)*, lies on <ANSWER>
- But: abundance & variety of Web data helps system to find an instance 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 <ANSWER> <Q_TERM_2>
- Does not use case
  - "What is a micron?"
  - "...a spokesman for Micron, a maker of semiconductors, said SIMMs are..."

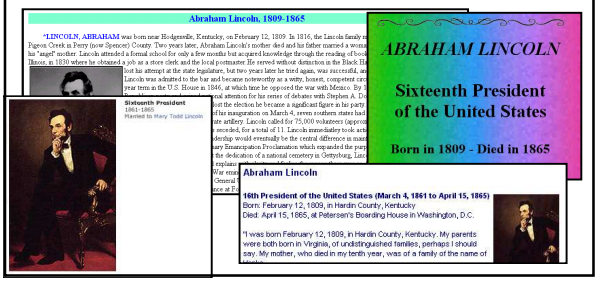
AskMSR

- ***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

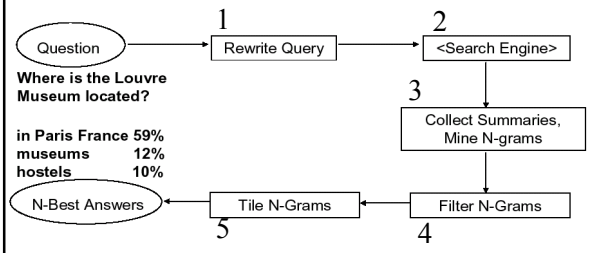


AskMSR: Shallow approach

- *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: 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"

  Nonsense, but who cares? It's only a few more queries

  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

+Louvre +Museum +located

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 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

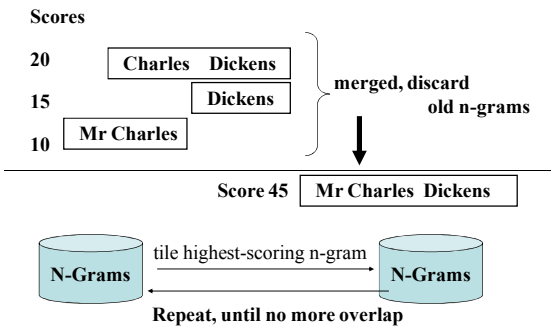
- Simple: Enumerate all N-grams (N=1,2,3 say) in all retrieved snippets
- Weight of an n-gram: occurrence count, each weighted by "reliability" (weight) of rewrite that fetched the document
- Example: "Who created the character of Scrooge?"
  - Dickens - 117
  - Christmas Carol - 78
  - Charles Dickens - 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… → **Date**
- Where… → **Location**
- 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



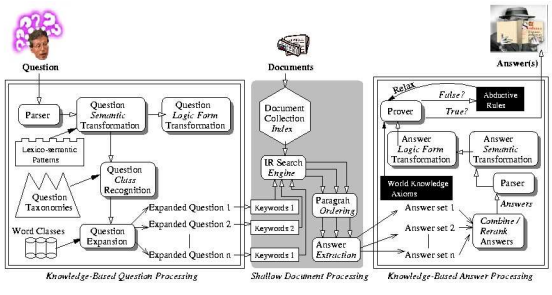
Results

- 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 redundancy 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

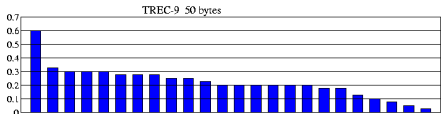
- In many scenarios (e.g., monitoring an individuals email…) we only have a small set of documents
- Works best/only for "Trivial Pursuit"-style fact-based questions
- Limited/brittle repertoire of
  - question categories
  - answer data types/filters
  - query rewriting rules

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



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'

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?*