Open Domain Question Answering

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu
Introduction

• Ad-hoc document retrieval solves a particular problem
  – Find relevant documents (quickly)
• Extracting useful information from the documents is a different task
  – Examples (in historical order):
    » Question answering (today)
    » Fact extraction (Sep 23)
    » Text data mining
    » Notice the differing complexity of these tasks…
Lecture Overview

• Introduction to Q/A
• Closed-domain Q/A
  – LUNAR
• Open-domain Q/A
  – Evaluation (TREC)
  – Indexing (Predictive Annotation)
  – Simple Pattern-Based Q/A
  – Common Problems and Solutions
  – Multi-strategy Approach
  – Evaluation Revisited
Introduction

Question:
When did the Titanic sink?

Answer:
April 15, 1912
(found in 11th document)
Introduction

The state of the art is not very advanced
Introduction to Question Answering

Goal: User types a question, system produces the correct answer
• Likely elements of a solution
  – Analyze question
  – Gather information
  – Distill answer(s)
  – Present answer(s)
• Several dimensions determine complexity and difficulty
  – Questions:
    » closed-domain (defined topic) vs. open-domain (any topic)
  – Data:
    » structured (e.g., relational) vs. unstructured (e.g., text)
  – Answers:
    » extracted (e.g., text snippet) vs. generated (e.g., dialog)
LUNAR was a system that answered questions about moon rocks and soil gathered by Apollo 11

• Architecture:
  – Parse English question into query in a formal query language
    » Syntactic analysis (rules, heuristics, semantics)
    » Semantic analysis (map parsed request into query language)
  – Run query on database to produce answer

• Resources:
  – Parser for a subset of English (size unclear)
  – Handled tense, modality, some anaphora, some conjunctions, ….
  – Vocabulary of about 3,500 words
LUNAR Example

• **Question:**
  – Do any samples have greater than 13 percent aluminum?

• **Query:**
  – (TEST (FOR SOME X1 / (SEQ SAMPLES) :
    T ;
    (CONTAIN
    X1
    (NPR* X2 / ‘AL203)
    (GREATERTHAN 13 PCT)))))

• **Answer:**
  – Yes
LUNAR Assessment

• **LUNAR system characteristics**
  – Closed domain (lunar geology and chemistry)
  – Structured data (information contained in a database)
  – Structured answers (information contained in database)
    » Avoids dialog problems
  – Sophisticated users demanding high accuracy

• **Labor intensive to build**
  – Complex system
  – High accuracy required
  – Few general-purpose resources available at the time
Closed-Domain Question Answering: Historical Perspective

• Research on systems like LUNAR continued for another decade
• Characteristics
  – Emphasis on syntactic parsing
  – Emphasis on domain knowledge
  – Emphasis on dialog management
  – Expensive to build
  – Brittle…prone to unexpected sudden failure
    » Especially at boundaries of domain knowledge
• A shift away from this style of NLP began in the late 1980s
  – Towards simpler, pattern-based techniques
Observations:

- **In a large text database (e.g., the Web)**
  - Some types of answers are relatively easy to find
    » Stereotypical text patterns
    » Repetition

- **Architecture for a simple open-domain Q/A system**
  - Analyze the question, produce an IR query
  - Retrieve a bunch of documents
  - Extract answer candidates (strings) from the documents
  - Select the best answer string(s)

- **First large-scale use in TREC-8 (1999)**
  - Now the predominant approach to question answering
Open-Domain Question Answering: Evaluation

Open-domain question answering evaluated at TREC (1999-2004)

• Task is influenced by what can be evaluated (with reasonable effort)

• Given a relatively unambiguous English question, find a fact-based answer
  – Harder than it sounds, e.g., “Where is the Taj Mahal?”
    » Agra, India (the expected answer)
    » New Jersey (a well-known Casino in Atlantic City)

• Answers expressed as (length limited) text fragments
  – E.g., “…at the city of Agra in the State of Uttar Pradesh, the…”
  – Lengths (over the years): 250 byte, 50 byte, exact
TREC Questions

• **Example questions:**
  – Who was the first American in space?
  – Where is the Taj Mahal?
  – How did Socrates die?
  – Why did David Koresh ask the FBI for a tape recorder?
  – Who is Colin Powell?
    » Many possible answers for this one – what’s correct?
  – What is Tyvek?

• **Open domain topics, but closed-class question form**
  – Questions conform to predictable language patterns
  – Most questions can be answered with little or no reasoning

(Czuba, 2002)
TREC Question Answering: Evaluation

- **Systems return up to N answers**
  - 5 in first three years, decreased to 1 in 2002
- **Human assessors judge answers**
  - Can accept multiple answers
    - E.g., “Agra”, “Uttar Pradesh”, “India”,
    - “Nil” is correct if answer not contained in the corpus
- **Systems scored on mean reciprocal rank of first correct answer**
  - 1 point if ranked first, ½ point if ranked 2\textsuperscript{nd}, 1/3 if ranked 3\textsuperscript{rd}, ….
  - 0 if none of the N answers are correct
  - Average across all questions
- **Number of questions answered correctly are also reported**

(Czuba, 2002)
Evaluation:
Issues and Problems

Problems with TREC evaluations
• **No penalties for answers that are correct but not helpful**
  – E.g., “Stuffing”: “…Agra, India, New Jersey, …”
• **No penalties for wrong answers**
  – In a real system, these might be confusing to a person
• **No reward for multiple, complementary answers**
• **No user model or user interaction**
  – So no guidance when question is ambiguous
• **Ambiguity about what is allowed**
  – Is it fair to find the answer on the Web, instead of the supplied corpus?
  – Is it fair to use a Gazetteer?

(Czuba, 2002)
Open-Domain Question Answering: Approaches

- **Indexing**
  - Predictive annotation
- **Simple pattern-based approach**
- **Common problems and solutions**
- **Multi-strategy approach**
Indexing with Predictive Annotation

• Some answers belong to well-defined semantic classes
  – People, places, monetary amounts, telephone numbers, …

• These may relatively easy to recognize at indexing time
  – And much harder to recognize at retrieval time
    » E.g., what query would retrieve Jamie’s telephone number?

• “Predictive annotation”: Index a document with “concepts” or “features” that are expected to be useful in (many) queries
  – E.g., people names, location names, phone numbers, etc.

• Add additional operators for use in queries
  – E.g., “Jamie Callan” NEAR/10 *TelephoneNumber
Predictive Annotation

In the early part of this century, the only means of transportation for travelers and mail between <LOCATION> Europe </LOCATION> and <LOCATION> North America </LOCATION> was by passenger steamship. By <DATE> 1907 </DATE>, the <COMPANY> Cunard Steamship Company </COMPANY> introduced the largest and fastest steamers in the <LOCATION> North Atlantic </LOCATION> service: the <NAME> Lusitania </NAME> and the <NAME> Mauritania </NAME>. Each had a gross tonnage of <WEIGHT> 31,000 tons </WEIGHT> and a maximum speed of <SPEED> 26 knots </SPEED>.

Predictive Annotation

• How is annotated text stored in the index?

In the early part of this century, the only means of transportation for travelers and mail between $LOCATION, Europe$ and $LOCATION North$ $LOCATION America$ was by passenger steamship. By $DATE 1907$, the $COMPANY, Cunard$ $COMPANY, Steamship$ $COMPANY, Company$ introduced the largest and fastest steamers in the $LOCATION, North$ $LOCATION, Atlantic$ service: the $NAME, Lusitania$ and the $NAME, Mauritania$. Each had a gross tonnage of $WEIGHT, 31,000$ $WEIGHT, tons$ and a maximum speed of $SPEED, 26$ $SPEED, knots$.

• Treat $QA-Token, term$ as meaning that $QA-Token and term$ occur at the same location in the text

• The query “$SPEED NEAR/1 knots” retrieves “26 knots”
Predictive Annotation

What makes a good QA-Token?

• Questions that would use the token
  – Can be recognized with high reliability (high Precision)
  – Occur frequently enough to be worth the effort

• The QA-Token can be recognized with high reliability in text
  – E.g., we have a high Precision organization recognizer

Common approaches to recognizing QA-Tokens

• Tables, lists, dictionaries
• Heuristics
• Hidden Markov Models
Indexing with Predictive Annotation

- **Advantages:**
  - Most computational cost is during indexing
    » Allows use of more sophisticated methods
  - Annotator has access to complete document text
    » Context may be important for recognizing some types of features

- **Disadvantages:**
  - Must know ahead of time which types of features/concepts are likely to be important
  - Can increase size of index considerably
    » E.g., by an order of magnitude if many features

- **Used (in varying amounts) by almost all open-domain Q/A systems**
Approach #1:
Simple Pattern-Based Question Answering

• There are many questions … but fewer types of questions
• Each type of question can be associated with:
  – Expectations about answer string characteristics
  – Strategies for retrieving documents that might have answers
  – Rules for identifying answer strings in documents
• Example: “Who is the President of CMU”
  – **Expectation:** Answer string contains a person name
    » Named-entity identification (covered later in course)
  – **Search Query:** “President, CMU, *PersonName”
  – **Rule:** “*PersonName, President of CMU”
    » Matches “…Jared Cohen, President of CMU…”
  – **Answer:** “Jared Cohen”
Pattern-Based Question Answering: Question Analysis

• **Input to question analysis:** Question

• **Question analysis:**
  – Identify named-entities
  – Categorize question
  – Match question-parts to templates

• **Result of question analysis:**
  – Search query
  – Answer expectations
  – Extraction strategy

• **Analysis patterns created manually (today)**
  – But, imagine a machine learning system in the future
Question Analysis

“Who is Elvis?”

• **Question type:** “Who”

• **Named-Entity Tagging:** “Who is <PersonName> Elvis </PersonName>

• **Name is given, so**
  - Search Query doesn’t need to contain a *PersonName* operator
  - Desired answer probably is a description
  - Likely extraction patterns:
    » “Elvis, the X” “…Elvis, the king of rock and roll…”
    » “the X Elvis” “…the legendary entertainer Elvis…”
    : : : : 

© 2004, Jamie Callan
Question Analysis

Frequency of question types on an Internet search engine

- 42% What
- 21% Where
- 20% Who
- 8% When
- 8% Why
- 2% Which
- 0% How

Relative difficulty of question types

- What is difficult
  - “What time…”
  - “What country…”
- Where is easy
- Who is easy
- When is easy
- Why is hard
- Which is hard
- How is hard
Example:
What is Jupiter?

**Question:** What is Jupiter?

1. What We Will Learn from Galileo
2. The Nature of Things: Jupiter's shockwaves - How a comet's bombardment has sparked activity on Earth
3. Jupiter-Bound Spacecraft Visits Earth on 6-Year Journey
4. STAR OF THE MAGI' THEORIES ECLIPSED?
5. Marketing & Media: Hearst, Burda to Scrap New Astrology Magazine
6. Greece, Italy Conflict On Cause Of Ship Crash That Kills 2, Injures 54
7. Interplanetary Spacecraft To `Visit' Earth With LaserGraphic
8. A List of Events During NASA's Galileo Mission to Jupiter
9. SHUTTLE ALOFT, SENDS GALILEO ON 6-YEAR VOYAGE TO JUPITER
10. Rebuilt Galileo Probe Readied For Long Voyage To Jupiter
Answer Extraction

• Select highly ranked sentences from highly ranked documents
  – “The planet Jupiter and its moons are in effect a mini-solar system, and Jupiter itself is often called a star that never caught fire.”

• Perform named-entity tagging (or extract from index) and perform part of speech tagging
  – “The/DT planet/NN <Location> Jupiter/NNP </Location> and/CC its/PRP moons/NNS are/VBP in/IN effect/NN a/DT mini-solar/JJ system/NN ./, and/CC <Location> Jupiter/NNP </Location> itself/PRP is/VBZ often/RB called/VBN a/DT star/NN that/IN never/RB caught/VBN fire/NN ./.”

• Apply extraction patterns
  – The/DT X Y, Y=Jupiter ➔ The planet Jupiter ➔ “planet”
Simple Pattern-Based Q/A: Assessment

• **Extremely effective when**
  – Question patterns are predictable
    » “Few” patterns cover the most likely questions
      • Could be several hundred
  – Not much variation in vocabulary
    » Simple word matching works
  – The corpus is huge (e.g., Web)
    » Odds of finding an answer document that matches the vocabulary and answer extraction rule improves

• **Somewhat labor intensive**
  – Patterns are created and tested manually
Common Problems and Solutions: Matching Questions to Answers

**Problem:** Document word order pattern isn’t exactly what was expected

**Solution:** “Soft matching” of answer patterns to document text

- **Use distance-based answer selection when no rule matches**
  - E.g., for “What is Jamie Callan’s telephone number?”
    - Use the telephone number nearest to the words “Jamie Callan”
    - Use the telephone number in the same sentence as “Jamie Callan”
Common Problems and Solutions: Matching Questions to Answers

**Problem:** Answer vocabulary doesn’t exactly match question vocabulary

**Solution:** Bridge the vocabulary mismatch

- **Use WordNet to identify simple relationships**
  - “Astronaut” is a type of “Person”
  - “Astronaut” and “Cosmonaut” are synonyms
Brief Digression: WordNet

• A lexical thesaurus organized into 4 taxonomies by part of speech
  – Created by George Miller & colleagues at Princeton University
• Inspired by psycholinguistic theories of human lexical memory.
• English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept.
• Different relations link the synonym sets.
  – Hyponyms: “…is a kind of X” relationships
  – Hypernyms: “X is a kind of …” relationships
  – Meronyms: “parts of X” relationships
WordNet:
5 Senses of “Plane”

<table>
<thead>
<tr>
<th>SynSet</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>airplane, aeroplane, plane</strong> -- (an aircraft that has fixed a wing and is powered by propellers or jets; &quot;the flight was delayed due to trouble with the airplane&quot;)</td>
<td></td>
</tr>
<tr>
<td>2. <strong>plane, sheet</strong> -- ((mathematics) an unbounded two-dimensional shape; &quot;we will refer to the plane of the graph as the X-Y plane&quot;; &quot;any line joining two points on a plane lies wholly on that plane&quot;)</td>
<td></td>
</tr>
<tr>
<td>3. <strong>plane</strong> -- (a level of existence or development; &quot;he lived on a worldly plane&quot;)</td>
<td></td>
</tr>
<tr>
<td>4. <strong>plane, planer, planing machine</strong> -- (a power tool for smoothing or shaping wood)</td>
<td></td>
</tr>
<tr>
<td>5. <strong>plane, carpenter's plane, woodworking plane</strong> -- (a carpenter's hand tool with an adjustable blade for smoothing or shaping wood; &quot;the cabinetmaker used a plane for the finish work&quot;)</td>
<td></td>
</tr>
</tbody>
</table>
WordNet: 
Hyponyms of “Airplane”

_airplane, aeroplane, plane_ -- (an aircraft that has fixed a wing and is powered by propellers or jets;  
=> _aircraft_ -- (a vehicle that can fly)  
=> _craft_ -- (a vehicle designed for navigation in or on water or air or through outer space)  
=> _vehicle_ -- (a conveyance that transports people or objects)  
=> _conveyance, transport_ -- (something that serves as a means of transportation)  
=> _instrumentality, instrumentation_ -- (an artifact (or system of artifacts) that is instrumental in accomplishing some end)  
=> _artifact, artefact_ -- (a man-made object)  
=> _object, physical object_ -- (a physical (tangible and visible) entity)  
=> _entity, something_ -- (anything having existence (living or nonliving))
WordNet: Hyponyms of “Airplane”

airplane, aeroplane, plane -- (an aircraft that has fixed a wing and is powered by propellers or jets; ....

=> airliner -- (an airplane that carries passengers)
=> amphibian, amphibious aircraft -- (designed to take off and land on water)
=> biplane -- (old fashioned; two wings one above the other)
=> bomber -- (a military aircraft that drops bombs during flight)
=> fighter, fighter aircraft, attack aircraft -- (a high-speed military or naval plane ...)
=> hangar queen -- (an airplane with a bad maintenance record)
=> jet, jet plane, jet-propelled plane -- (an airplane powered by ... jet engines)
=> monoplane -- (an airplane with a single wing)
=> propeller plane -- (an airplane that is driven by a propeller)
=> seaplane, hydroplane -- (an airplane that can land on or take off from water)
=> ski-plane -- (a plane equipped with skis so it can land on a snowfield)
=> turbojet, turboprop -- (airplane powered by a turbojet engine)
WordNet: Meronyms of “Airplane”

airplane, aeroplane, plane -- (an aircraft that has fixed a wing …

HAS PART: accelerator, accelerator pedal, gas pedal, gas, throttle, gun -- ….  
HAS PART: cowl, cowling, hood -- (metal part that covers the engine)  
HAS PART: escape hatch -- (a means of escape in an emergency)  
HAS PART: fuselage -- (the central portion of an airplane …  
HAS PART: landing gear -- (an undercarriage that supports the weight ….  
HAS PART: pod, fuel pod -- (a detachable container of fuel on an airplane)  
HAS PART: radome, radar dome -- (a housing for a radar antenna; ….  
HAS PART: windshield, windsreen -- (transparent (as of glass) to protect …  
HAS PART: wing -- (one of the horizontal airfoils on either side ….  

35
Common Problems and Solutions: Improving the Set of Retrieved Documents

**Problem:** Sometimes the IR system can’t find any documents that have answers (even though the documents are in the corpus)

**Solution:** Get a broader set of documents

- **Two (common) approaches**
  - If answer extractor fails to find an answer, kick the question back to the search engine with instructions to widen the search
    - Assumes answer extractors can tell when they fail
  - Use a variety of retrieval strategies to retrieve documents
    - E.g., all words within one sentence, then all words within one paragraph, then within same document, …
    - E.g., add synonyms to query or do query expansion
    - Simple, but much higher computational expense
Common Problems and Solutions: Improving the Set of Retrieved Documents

**Problem:** Word sequence patterns have limited power

**Solution:** Create patterns that use syntactic information

- **Partial syntactic parsing of documents**
  - Is this noun the subject or the object of the sentence?

- **Allows more complex patterns**
  - Question: “Who shot Kennedy?”
  - “Who” implies a person that should be subject of answer
  - “Kennedy” should be direct object of answer
  - Pattern: `<subject>` shot Kennedy
  - Matching text: “Oswald shot Kennedy”
Common Problems and Solutions: Improving the Set of Retrieved Documents

**Problem:** Multiple answer candidates

**Solutions:**

- **Features used to represent answer candidates**
  - Frequency
  - Distance to question words
  - Location in answer passage(s)
  - Answer passage score(s)

- **Selection functions**
  - Created manually
  - Learned from training data
Multi-Strategy Approach

- **The state-of-the-art is the SMU/LCC Falcon system**
  - Informed use of standard IR
  - Use of broad ontology (extended WordNet)
  - Heavy on NLP
  - Answer verification

- **Similar to most other systems in general architecture, but…**
  - Much more careful tuning
    » Of algorithms
    » Of resources
  - More sophisticated control of IR and NLP
  - Feedback loops

(Czuba, 2002)
Multi-Strategy Approach: Question Analysis

**Question analysis produces:**

- **Parsing and named-entity recognition**
  - Exceptions for “special cases”

- **Expected answer type**
  - Determined using question focus
    - E.g., “What city is the Logan airport in?”
  - Expected answer type determined by looking up words in WordNet

- **Question logical form**

- **Question keywords and query formulation**
Expected Answer Types

(M. Pasca and S. Harabagiu, SIGIR 2001)
Multi-Strategy Approach: Expected Answer Type

- **Drives processing of paragraphs**
  - Passages need to contain the expected answer type

- **Answer types are mapped to named-entity categories that can be recognized in text**
  - Many to many mapping
  - Based on ontology, robust

![Figure 2: Many-to-many mappings between named entities and tops of the Answer Type Taxonomy](M. Pasca and S. Harabagiu, SIGIR 2001)
Paragraph Retrieval

- **Boolean retrieval with loops**
  - Different from multiple queries…only uses additional queries when necessary
  - Fewer candidates for analysis components to consider
- **Loop 1: Query keyword loop**
  - Keywords added/dropped to make query more/less specific
- **Loop 2: Keyword alternations**
  - Try morphological variants and synonyms
- **Loop 3: Semantic alternations**
  - Try semantic alternatives
Multi-Strategy Approach: Feedback Loops

Figure 3: Retrieval Feedbacks in a Q/A System
(M. Pasca and S. Harabagiu, SIGIR 2001)
Answer Verification

- Parse passages to create a dependency tree among words
- Attempt to unify logical forms of question and answer text

(M. Pasca and S. Harabagiu, SIGIR 2001)
The LCC System in 2003

(Harabagiu, et al, 2004)
The LCC System in 2003

• Lists are handled by
  – the usual factoid mechanisms
  – a question-specific threshold
    » Based on similarities of answer contexts

• Definition questions are matched with patterns
  – “What is <Question-Phrase>?” ➔
    “<Question-Phrase>, which means <Answer-Phrase>”
  – “What is Iqra?” matches “Iqra, which means read in Arabic, …”
  – 38 patterns for definition questions
    » <QP>, [a, an, the] <AP>
    » <AP> such as <QP>

(Harabagiu, et al, 2004)
The LCC System in 2003

• Out of 289 factoid questions answered correctly, 234 were done by answer extraction
  – 71% based on name recognition
  – 29% based on concepts and answer type taxonomy

• Very detailed categories
  – E.g., “manner of death” category
  – Start with a set of seed patterns…
    …learn a much larger set of “manner of death” patterns…
    …and manually verify them
    » Based on Ellen Riloff’s work

• 65 correct factoid answers due to theorem prover (23%)

(Harabagiu, et al, 2004)
The LCC System in 2003

- **Accuracy:**
  - 70% on factoid questions (!!!)
  - 39% on list questions
  - 36% (exact) and 44% (sentence) on definition questions
  - Overall: 54% and 56%

(Harabagiu, et al, 2004)
Multi-Strategy Approach: Assessment

• **Strengths**
  – Controlled use of IR system
    » Query expansion via lexical and semantic equivalents
    » Believed to be the major power of the system
  – Tailored resources (not discussed today)
    » WordNet, parser, etc
  – Answer verification
    » Initially thought to be the key component of the system
    » Now…not so sure

• **Weaknesses**
  – Complex system, contribution of each component unclear
TREC 2000 Q/A Evaluation

About 979,000 newswire documents from TREC CDs 1, 2, 3; 693 questions; 50 byte limit for answers
TREC 2001 Q/A Evaluation

About 979,000 newswire documents from TREC CDs 1, 2, 3;
500 questions; 50 byte limit for answers; NIL if no answer in corpus
TREC 2002 Q/A Evaluation

About 1 million documents from NY Times, AP, Xinhua News Agency,

500 questions
One exact answer, or NIL
Rank answers by confidence

Main Task Results

(E. Voorhees, 2002)
TREC 2003 Q/A Evaluation

About 1 million documents from NY Times, AP, Xinhua News Agency,
Different question types

QA Main Task Results

Final combined scores for best main task run per group for top 10 groups

Text REtrieval Conference (TREC)
(E. Voorhees, 2003)
Question Answering Summary

- Significant progress over the years on factoid questions
  - Due to careful analysis of question types
  - Due to large numbers of patterns for extracting answers
  - Answer justification for the hard or non-obvious cases
- This is hardly deep NLP
  - But, perhaps it is useful NLP
  - Most questions are simple, most answers are easy to find
Is the TREC Q/A task realistic?
• It depends on the task you’re trying to model
• Much is missing from the TREC task
  – Domain knowledge
  – Inference
  – Complex documents
  – Dialog
  – …
• But...the real question is what Q/A task requires these?
Lecture Overview

• **Introduction to Q/A**
• **Closed-domain Q/A**
  – LUNAR
• **Open-domain Q/A**
  – Evaluation (TREC)
  – Indexing (Predictive Annotation)
  – Simple Pattern-Based Q/A
  – Common Problems and Solutions
  – Multi-strategy Approach
  – Evaluation Revisited
For More Information


• R. Srihari and W. Li. “Information extraction supported question answering.” In TREC-8 conference proceedings. Available at http://trec.nist.gov/pubs/trec8/t8_proceedings.html