Question-Answering: Shallow & Deep Techniques for NLP

Ling571
Deep Processing Techniques for NLP
March 9, 2011

Examples from Dan Jurafsky)
Roadmap

- Question-Answering:
  - Definitions & Motivation

- Basic pipeline:
  - Question processing
  - Retrieval
  - Answering processing

- Shallow processing: AskMSR (Brill)

- Deep processing: LCC (Moldovan, Harabagiu, et al)

- Wrap-up
Why QA?

- Grew out of information retrieval community
- Web search is great, but...
  - Sometimes you don’t just want a ranked list of documents
  - Want an answer to a question!
    - Short answer, possibly with supporting context
Why QA?

- Grew out of information retrieval community
- Web search is great, but...
  - Sometimes you don’t just want a ranked list of documents
  - Want an answer to a question!
    - Short answer, possibly with supporting context
- People ask questions on the web
  - Web logs:
    - Which English translation of the bible is used in official Catholic liturgies?
    - Who invented surf music?
    - What are the seven wonders of the world?
Why QA?

- Grew out of information retrieval community
- Web search is great, but...
  - Sometimes you don’t just want a ranked list of documents
  - Want an answer to a question!
    - Short answer, possibly with supporting context
- People ask questions on the web
  - Web logs:
    - Which English translation of the bible is used in official Catholic liturgies?
    - Who invented surf music?
    - What are the seven wonders of the world?
  - Account for 12-15% of web log queries
Search Engines and Questions

- What do search engines do with questions?
Search Engines and Questions

- What do search engines do with questions?
  - Often remove ‘stop words’
    - Invented surf music/seven wonders world/....
  - Not a question any more, just key word retrieval

- How well does this work?
Search Engines and Questions

- What do search engines do with questions?
  - Often remove ‘stop words’
    - Invented surf music/seven wonders world/....
  - Not a question any more, just key word retrieval

- How well does this work?
  - *Who invented surf music?*
Search Engines and Questions

- What do search engines do with questions?
  - Often remove ‘stop words’
    - Invented surf music/seven wonders world/....
  - Not a question any more, just key word retrieval

- How well does this work?
  - *Who invented surf music?*
    - Rank #2 snippet:
      - Dick Dale *invented surf music*
    - Pretty good, but...
Search Engines & QA

- Who was the prime minister of Australia during the Great Depression?
Search Engines & QA

- Who was the prime minister of Australia during the Great Depression?
- Rank 1 snippet:
  - The conservative Prime Minister of Australia, Stanley Bruce
Search Engines & QA

- **Who was the prime minister of Australia during the Great Depression?**
  - Rank 1 snippet:
    - The conservative *Prime Minister of Australia*, Stanley Bruce
  - Wrong!
    - Voted out just before the Depression

- **What is the total population of the ten largest capitals in the US?**
Search Engines & QA

- **Who was the prime minister of Australia during the Great Depression?**
  - Rank 1 snippet:
    - The conservative *Prime Minister of Australia*, Stanley Bruce
  - Wrong!
    - Voted out just before the Depression

- **What is the total population of the ten largest capitals in the US?**
  - Rank 1 snippet:
    - The table below lists the *largest 50 cities in the United States*
      .....
Search Engines & QA

- **Who was the prime minister of Australia during the Great Depression?**
  - Rank 1 snippet:
    - The conservative Prime Minister of Australia, Stanley Bruce
  - Wrong!
    - Voted out just before the Depression

- **What is the total population of the ten largest capitals in the US?**
  - Rank 1 snippet:
    - The table below lists the largest 50 cities in the United States ..... 
  - The answer is in the document – with a calculator.
Search Engines and QA
Search Engines and QA

- Search for exact question string
  - “Do I need a visa to go to Japan?”
    - Result: Exact match on Yahoo! Answers

- Find ‘Best Answer’ and return following chunk
Search Engines and QA

- Search for exact question string
  - “Do I need a visa to go to Japan?”
    - Result: Exact match on Yahoo! Answers

- Find ‘Best Answer’ and return following chunk

- Works great if the question matches exactly
  - Many websites are building archives
  - What if it doesn’t match?
Search Engines and QA

- Search for exact question string
  - “Do I need a visa to go to Japan?”
    - Result: Exact match on Yahoo! Answers
  - Find ‘Best Answer’ and return following chunk

- Works great if the question matches exactly
  - Many websites are building archives
  - What if it doesn’t match?
    - ‘Question mining’ tries to learn paraphrases of questions to get answers
Perspectives on QA

- TREC QA track (~2000-)
  - Initially pure factoid questions, with fixed length answers
    - Based on large collection of fixed documents (news)
    - Increasing complexity: definitions, biographical info, etc
      - Single response
Perspectives on QA

- TREC QA track (~2000---)
  - Initially pure factoid questions, with fixed length answers
    - Based on large collection of fixed documents (news)
    - Increasing complexity: definitions, biographical info, etc
      - Single response

- Reading comprehension (Hirschman et al, 2000---)
  - Think SAT/GRE
    - Short text or article (usually middle school level)
    - Answer questions based on text
  - Also, ‘machine reading’
Perspectives on QA

- TREC QA track (~2000---)
  - Initially pure factoid questions, with fixed length answers
    - Based on large collection of fixed documents (news)
    - Increasing complexity: definitions, biographical info, etc
      - Single response
  - Reading comprehension (Hirschman et al, 2000---)
    - Think SAT/GRE
      - Short text or article (usually middle school level)
      - Answer questions based on text
    - Also, ‘machine reading’

- And, of course, Jeopardy! and Watson
## Question Answering (a la TREC)

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where is the Louvre Museum located?</td>
<td>in Paris, France</td>
</tr>
<tr>
<td>What’s the abbreviation for limited partnership?</td>
<td>L.P.</td>
</tr>
<tr>
<td>What are the names of Odin’s ravens?</td>
<td>Huginn and Muninn</td>
</tr>
<tr>
<td>What currency is used in China?</td>
<td>the yuan</td>
</tr>
<tr>
<td>What kind of nuts are used in marzipan?</td>
<td>almonds</td>
</tr>
<tr>
<td>What instrument does Max Roach play?</td>
<td>drums</td>
</tr>
<tr>
<td>What’s the official language of Algeria?</td>
<td>Arabic</td>
</tr>
<tr>
<td>What is the telephone number for the University of Colorado, Boulder?</td>
<td>(303)492-1411</td>
</tr>
<tr>
<td>How many pounds are there in a stone?</td>
<td>14</td>
</tr>
</tbody>
</table>
Basic Strategy

- Given an indexed document collection, and
- A question:
- Execute the following steps:
  - Query formulation
  - Question classification
  - Passage retrieval
  - Answer processing
  - Evaluation
Query Formulation

- Convert question suitable form for IR
- Strategy depends on document collection
  - Web (or similar large collection):
Query Formulation

- Convert question suitable form for IR
- Strategy depends on document collection
  - Web (or similar large collection):
    - ‘stop structure’ removal:
      - Delete function words, q-words, even low content verbs
  - Corporate sites (or similar smaller collection):
Query Formulation

- Convert question suitable form for IR
- Strategy depends on document collection
  - Web (or similar large collection):
    - ‘stop structure’ removal:
      - Delete function words, q-words, even low content verbs
  - Corporate sites (or similar smaller collection):
    - Query expansion
      - Can’t count on document diversity to recover word variation
Query Formulation

- Convert question suitable form for IR
- Strategy depends on document collection
  - Web (or similar large collection):
    - ‘stop structure’ removal:
      - Delete function words, q-words, even low content verbs
  - Corporate sites (or similar smaller collection):
    - Query expansion
      - Can’t count on document diversity to recover word variation
      - Add morphological variants, WordNet as thesaurus
Query Formulation

- Convert question suitable form for IR
- Strategy depends on document collection
  - Web (or similar large collection):
    - ‘stop structure’ removal:
      - Delete function words, q-words, even low content verbs
  - Corporate sites (or similar smaller collection):
    - Query expansion
      - Can’t count on document diversity to recover word variation
      - Add morphological variants, WordNet as thesaurus
      - Reformulate as declarative: rule-based
        - Where is X located -> X is located in
Question Classification

- Answer type recognition
  - Who
Question Classification

- Answer type recognition
  - Who -> Person
  - What Canadian city ->
Question Classification

- Answer type recognition
  - Who -> Person
  - What Canadian city -> City
  - What is surf music -> Definition

- Identifies type of entity (e.g. Named Entity) or form (biography, definition) to return as answer
Question Classification

- Answer type recognition
  - Who -> Person
  - What Canadian city -> City
  - What is surf music -> Definition

- Identifies type of entity (e.g. Named Entity) or form (biography, definition) to return as answer
  - Build ontology of answer types (by hand)

- Train classifiers to recognize
Question Classification

- Answer type recognition
  - Who -> Person
  - What Canadian city -> City
  - What is surf music -> Definition

- Identifies type of entity (e.g. Named Entity) or form (biography, definition) to return as answer
  - Build ontology of answer types (by hand)

- Train classifiers to recognize
  - Using POS, NE, words
Question Classification

- Answer type recognition
  - Who -> Person
  - What Canadian city -> City
  - What is surf music -> Definition

- Identifies type of entity (e.g. Named Entity) or form (biography, definition) to return as answer
  - Build ontology of answer types (by hand)

- Train classifiers to recognize
  - Using POS, NE, words
  - Synsets, hyper/hypo-nyms
<table>
<thead>
<tr>
<th>Tag</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABBREVIATION</td>
<td></td>
</tr>
<tr>
<td>abb</td>
<td>What’s the abbreviation for limited partnership?</td>
</tr>
<tr>
<td>exp</td>
<td>What does the “e” stand for in the equation E=mc2?</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>definition</td>
<td>What are tammins?</td>
</tr>
<tr>
<td>description</td>
<td>What are the words to the Canadian National anthem?</td>
</tr>
<tr>
<td>manner</td>
<td>How can you get rust stains out of clothing?</td>
</tr>
<tr>
<td>reason</td>
<td>What caused the Titanic to sink?</td>
</tr>
<tr>
<td>ENTITY</td>
<td></td>
</tr>
<tr>
<td>animal</td>
<td>What are the names of Odin’s ravens?</td>
</tr>
<tr>
<td>body</td>
<td>What part of your body contains the corpus callosum?</td>
</tr>
<tr>
<td>color</td>
<td>What colors make up a rainbow?</td>
</tr>
<tr>
<td>creative</td>
<td>In what book can I find the story of Aladdin?</td>
</tr>
<tr>
<td>currency</td>
<td>What currency is used in China?</td>
</tr>
<tr>
<td>disease/medicine</td>
<td>What does Salk vaccine prevent?</td>
</tr>
<tr>
<td>event</td>
<td>What war involved the battle of Chapultepec?</td>
</tr>
<tr>
<td>food</td>
<td>What kind of nuts are used in marzipan?</td>
</tr>
<tr>
<td>instrument</td>
<td>What instrument does Max Roach play?</td>
</tr>
<tr>
<td>lang</td>
<td>What’s the official language of Algeria?</td>
</tr>
<tr>
<td>letter</td>
<td>What letter appears on the cold-water tap in Spain?</td>
</tr>
<tr>
<td>other</td>
<td>What is the name of King Arthur’s sword?</td>
</tr>
<tr>
<td>plant</td>
<td>What are some fragrant white climbing roses?</td>
</tr>
<tr>
<td>product</td>
<td>What is the fastest computer?</td>
</tr>
<tr>
<td>religion</td>
<td>What religion has the most members?</td>
</tr>
<tr>
<td>sport</td>
<td>What was the name of the ball game played by the Mayans?</td>
</tr>
<tr>
<td>substance</td>
<td>What fuel do airplanes use?</td>
</tr>
<tr>
<td>symbol</td>
<td>What is the chemical symbol for nitrogen?</td>
</tr>
<tr>
<td>technique</td>
<td>What is the best way to remove wallpaper?</td>
</tr>
<tr>
<td>term</td>
<td>How do you say “Grandma” in Irish?</td>
</tr>
<tr>
<td>vehicle</td>
<td>What was the name of Captain Bligh’s ship?</td>
</tr>
<tr>
<td>word</td>
<td>What’s the singular of dice?</td>
</tr>
<tr>
<td>HUMAN</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>Who was Confucius?</td>
</tr>
<tr>
<td>group</td>
<td></td>
</tr>
<tr>
<td>ind</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td></td>
</tr>
<tr>
<td>LOCATION</td>
<td>city</td>
</tr>
<tr>
<td></td>
<td>country</td>
</tr>
<tr>
<td></td>
<td>mountain</td>
</tr>
<tr>
<td></td>
<td>other</td>
</tr>
<tr>
<td></td>
<td>state</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>code</td>
</tr>
<tr>
<td></td>
<td>count</td>
</tr>
<tr>
<td></td>
<td>date</td>
</tr>
<tr>
<td></td>
<td>distance</td>
</tr>
<tr>
<td></td>
<td>money</td>
</tr>
<tr>
<td></td>
<td>order</td>
</tr>
<tr>
<td></td>
<td>other</td>
</tr>
<tr>
<td></td>
<td>period</td>
</tr>
<tr>
<td></td>
<td>percent</td>
</tr>
<tr>
<td></td>
<td>speed</td>
</tr>
<tr>
<td></td>
<td>temp</td>
</tr>
<tr>
<td></td>
<td>size</td>
</tr>
<tr>
<td></td>
<td>weight</td>
</tr>
</tbody>
</table>
Passage Retrieval

- Why not just perform general information retrieval?
Passage Retrieval

- Why not just perform general information retrieval?
  - Documents too big, non-specific for answers

- Identify shorter, focused spans (e.g., sentences)
Passage Retrieval

- Why not just perform general information retrieval?
  - Documents too big, non-specific for answers

- Identify shorter, focused spans (e.g., sentences)
  - Filter for correct type: answer type classification
  - Rank passages based on a trained classifier
    - Features:
      - Question keywords, Named Entities
      - Longest overlapping sequence,
      - Shortest keyword-covering span
      - N-gram overlap b/t question and passage
Passage Retrieval

- Why not just perform general information retrieval?
  - Documents too big, non-specific for answers

- Identify shorter, focused spans (e.g., sentences)
  - Filter for correct type: answer type classification
  - Rank passages based on a trained classifier
    - Features:
      - Question keywords, Named Entities
      - Longest overlapping sequence,
      - Shortest keyword-covering span
      - N-gram overlap b/t question and passage

- For web search, use result snippets
Answer Processing

- Find the specific answer in the passage
Answer Processing

- Find the specific answer in the passage

- Pattern extraction-based:
  - Include answer types, regular expressions

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textless AP\textgreater such as \textless QP\textgreater</td>
<td>What is autism?</td>
<td>“, developmental disorders such as autism”</td>
</tr>
<tr>
<td>\textless QP\textgreater, a \textless AP\textgreater</td>
<td>What is a caldera?</td>
<td>“the Long Valley caldera, a \textit{volcanic crater} 19 miles long”</td>
</tr>
</tbody>
</table>

- Similar to relation extraction:
  - Learn relation b/t answer type and aspect of question
Answer Processing

- Find the specific answer in the passage
- Pattern extraction-based:
  - Include answer types, regular expressions

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;AP&gt; such as &lt;QP&gt;</td>
<td>What is autism?</td>
<td>“developmental disorders such as autism”</td>
</tr>
<tr>
<td>&lt;QP&gt;, a &lt;AP&gt;</td>
<td>What is a caldera?</td>
<td>“the Long Valley caldera, a volcanic crater 19 miles long”</td>
</tr>
</tbody>
</table>

- Similar to relation extraction:
  - Learn relation b/t answer type and aspect of question
  - E.g. date-of-birth/person name; term/definition
  - Can use bootstrap strategy for contexts, like Yarowsky
  - <NAME> (<BD>-<DD>) or <NAME> was born on <BD>
Answer Processing

- Find the specific answer in the passage
- Pattern extraction-based:
  - Include answer types, regular expressions

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;AP&gt; such as &lt;QP&gt;</code></td>
<td>What is autism?</td>
<td>“developmental disorders such as autism”</td>
</tr>
<tr>
<td><code>&lt;QP&gt;, a &lt;AP&gt;</code></td>
<td>What is a caldera?</td>
<td>“the Long Valley caldera, a <strong>volcanic crater</strong> 19 miles long”</td>
</tr>
</tbody>
</table>

- Similar to relation extraction:
  - Learn relation b/t answer type and aspect of question
    - E.g. date-of-birth/person name; term/definition
    - Can use bootstrap strategy for contexts, like Yarowsky
  - `<NAME>` (<BD>-<DD>) or `<NAME>` was born on `<BD>`
Evaluation

- Classical:
  - Return ranked list of answer candidates
Evaluation

- Classical:
  - Return ranked list of answer candidates
  - Idea: Correct answer higher in list => higher score

- Measure: Mean Reciprocal Rank (MRR)
Evaluation

• Classical:
  • Return ranked list of answer candidates
  • Idea: Correct answer higher in list => higher score

• Measure: Mean Reciprocal Rank (MRR)
  • For each question,
    • Get reciprocal of rank of first correct answer
    • E.g. correct answer is 4 => \( \frac{1}{4} \)
    • None correct => 0
  • Average over all questions

\[
MRR = \frac{\sum_{i=1}^{N} \frac{1}{\text{rank}_i}}{N}
\]
AskMSR

- Shallow Processing for QA

Diagram:
1. Rewrite Query
   - Question: Where is the Louvre Museum located?
   - Query: "+the Louvre Museum +is located"

2. Search Engine

3. Collect Summaries, Mine N-grams

4. Filter N-grams

5. N-Best Answers

Statistics:
- In Paris France: 59%
- Museums: 12%
- Hostels: 10%
Intuition

- Redundancy is useful!
  - If similar strings appear in many candidate answers, likely to be solution
    - Even if can’t find obvious answer strings
Intuition

- Redundancy is useful!
  - If similar strings appear in many candidate answers, likely to be solution
    - Even if can’t find obvious answer strings

- Q: How many times did Bjorn Borg win Wimbledon?
  - Bjorn Borg blah blah blah Wimbledon blah 5 blah
  - Wimbledon blah blah blah Bjorn Borg blah 37 blah.
  - blah Bjorn Borg blah blah 5 blah blah Wimbledon
  - 5 blah blah Wimbledon blah blah blah Bjorn Borg.
Intuition

- Redundancy is useful!
  - If similar strings appear in many candidate answers, likely to be solution
    - Even if can’t find obvious answer strings

- Q: How many times did Bjorn Borg win Wimbledon?
  - Bjorn Borg blah blah blah Wimbledon blah 5 blah
  - Wimbledon blah blah blah Bjorn Borg blah 37 blah.
  - blah Bjorn Borg blah blah 5 blah blah Wimbledon
  - 5 blah blah Wimbledon blah blah blah Bjorn Borg.
  - Probably 5
Query Reformulation

- Identify question type:
  - E.g. Who, When, Where,…

- Create question-type specific rewrite rules:
Query Reformulation

- Identify question type:
  - E.g. Who, When, Where,…

- Create question-type specific rewrite rules:
  - Hypothesis: Wording of question similar to answer
    - For ‘where’ queries, move ‘is’ to all possible positions
      - Where is the Louvre Museum located? =>
        - Is the Louvre Museum located
        - The is Louvre Museum located
        - The Louvre Museum is located, .etc.
Query Reformulation

- Identify question type:
  - E.g. Who, When, Where,...

- Create question-type specific rewrite rules:
  - Hypothesis: Wording of question similar to answer
    - For ‘where’ queries, move ‘is’ to all possible positions
      - Where is the Louvre Museum located? =>
        - Is the Louvre Museum located
        - The is Louvre Museum located
        - The Louvre Museum is located, .etc.

- Create type-specific answer type (Person, Date, Loc)
Query Reformulation

- Shallow processing:
  - No parsing, only POS tagging
  - Only 10 rewrite types
Query Reformulation

- Shallow processing:
  - No parsing, only POS tagging
  - Only 10 rewrite types

- Issue:
  - Some patterns more reliable than others
  - Weight by reliability
    - Precision/specificity – manually assigned
Retrieval, N-gram Mining & Filtering

- Run reformulated queries through search engine
  - Collect (lots of) result snippets
Retrieval, N-gram Mining & Filtering

- Run reformulated queries through search engine
  - Collect (lots of) result snippets

- Collect all uni-, bi-, and tri-grams from snippets
Retrieval, N-gram Mining & Filtering

- Run reformulated queries through search engine
  - Collect (lots of) result snippets

- Collect all uni-, bi-, and tri-grams from snippets

- Weight each n-gram by
  - Sum over # of occurrences: Query_form_weight
Retrieval, N-gram Mining & Filtering

- Run reformulated queries through search engine
  - Collect (lots of) result snippets

- Collect all uni-, bi-, and tri-grams from snippets

- Weight each n-gram by
  - Sum over # of occurrences: Query_form_weight

- Filter/reweight n-grams by match with answer type
  - Hand-crafted rules
N-gram Tiling

• Concatenates N-grams into longer answers
  • Greedy method:
    • Select highest scoring candidate, try to add on others
    • Add best concatenation, remove lowest
    • Repeat until no overlap
N-gram Tiling

- Concatenates N-grams into longer answers
- Greedy method:
  - Select highest scoring candidate, try to add on others
  - Add best concatenation, remove lowest
  - Repeat until no overlap

Scores

- 20 Charles Dickens
- 15 Dickens
- 10 Mr Charles

Score 45 Mr Charles Dickens

merged, discard old n-grams
Deep Processing Technique for QA

- LCC (Moldovan, Harabagiu, et al)
Deep Processing: Query/Answer Formulation

- Preliminary shallow processing:
  - Tokenization, POS tagging, NE recognition, Preprocess

- Parsing creates syntactic representation:
  - Focused on nouns, verbs, and particles
    - Attachment

- Coreference resolution links entity references

- Translate to full logical form
  - As close as possible to syntax
Syntax to Logical Form
Syntax to Logical Form
Syntax to Logical Form
Deep Processing: Answer Selection

- Lexical chains:
  - Bridge gap in lexical choice b/t Q and A
    - Improve retrieval and answer selection
Deep Processing: Answer Selection

- Lexical chains:
  - Bridge gap in lexical choice b/t Q and A
    - Improve retrieval and answer selection
  - Create connections between synsets through topicality
    - Q: When was the internal combustion engine invented?
      - A: The first internal-combustion engine was built in 1867.
    - invent → create_mentally → create → build
Deep Processing: Answer Selection

- **Lexical chains:**
  - Bridge gap in lexical choice b/t Q and A
    - Improve retrieval and answer selection
  - Create connections between synsets through topicality
    - **Q:** When was the internal combustion engine invented?
    - **A:** The first internal-combustion engine was built in 1867.
    - `invent` → `create_mentally` → `create` → `build`

- **Perform abductive reasoning**
  - Tries to justify answer given question
Deep Processing: Answer Selection

- Lexical chains:
  - Bridge gap in lexical choice b/t Q and A
    - Improve retrieval and answer selection
  - Create connections between synsets through topicality
    - \( Q: \text{When was the internal combustion engine invented?} \)
    - \( A: \text{The first internal-combustion engine was built in 1867.} \)
    - \( \text{invent} \rightarrow \text{create}_\text{mentally} \rightarrow \text{create} \rightarrow \text{build} \)

- Perform abductive reasoning
  - Tries to justify answer given question
  - Yields 30% improvement in accuracy!
Question Answering Example

- How hot does the inside of an active volcano get?
- \texttt{get(TEMPERATURE, inside(volcano(\text{active})))}
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
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’
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’
A Victory for Deep Processing

<table>
<thead>
<tr>
<th>Run Tag</th>
<th>Score</th>
<th>#</th>
<th>%</th>
<th>Inexact</th>
<th>Prec</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCCmain2002</td>
<td>0.856</td>
<td>415</td>
<td>83.0</td>
<td>8</td>
<td>0.578</td>
<td>0.804</td>
</tr>
<tr>
<td>exactanswer</td>
<td>0.691</td>
<td>271</td>
<td>54.2</td>
<td>12</td>
<td>0.222</td>
<td>0.848</td>
</tr>
<tr>
<td>pris2002</td>
<td>0.610</td>
<td>290</td>
<td>58.0</td>
<td>17</td>
<td>0.241</td>
<td>0.891</td>
</tr>
<tr>
<td>IRST02D1</td>
<td>0.589</td>
<td>192</td>
<td>38.4</td>
<td>17</td>
<td>0.167</td>
<td>0.217</td>
</tr>
<tr>
<td>IBMPQSQACYC</td>
<td>0.588</td>
<td>179</td>
<td>35.8</td>
<td>9</td>
<td>0.196</td>
<td>0.630</td>
</tr>
<tr>
<td>uwmtB3</td>
<td>0.512</td>
<td>184</td>
<td>36.8</td>
<td>20</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>BBN2002C</td>
<td>0.499</td>
<td>142</td>
<td>28.4</td>
<td>18</td>
<td>0.182</td>
<td>0.087</td>
</tr>
<tr>
<td>isi02</td>
<td>0.498</td>
<td>149</td>
<td>29.8</td>
<td>15</td>
<td>0.385</td>
<td>0.109</td>
</tr>
<tr>
<td>limsiQalir2</td>
<td>0.497</td>
<td>133</td>
<td>26.6</td>
<td>11</td>
<td>0.188</td>
<td>0.196</td>
</tr>
<tr>
<td>ali2002b</td>
<td>0.496</td>
<td>181</td>
<td>36.2</td>
<td>15</td>
<td>0.156</td>
<td>0.848</td>
</tr>
<tr>
<td>ibmsqa02c</td>
<td>0.455</td>
<td>145</td>
<td>29.0</td>
<td>44</td>
<td>0.224</td>
<td>0.239</td>
</tr>
<tr>
<td>FDUT11QA1</td>
<td>0.434</td>
<td>124</td>
<td>24.8</td>
<td>6</td>
<td>0.139</td>
<td>0.957</td>
</tr>
<tr>
<td>aranea02a</td>
<td>0.433</td>
<td>152</td>
<td>30.4</td>
<td>36</td>
<td>0.235</td>
<td>0.174</td>
</tr>
<tr>
<td>nuslamp2002</td>
<td>0.396</td>
<td>105</td>
<td>21.0</td>
<td>17</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

AskMSR: 0.24 on TREC data; 0.42 on TREC queries w/full web
Conclusions

- Deep processing for QA
  - Exploits parsing, semantics, anaphora, reasoning
  - Computationally expensive
    - But tractable because applied only to
      - Questions and Passages

- Trends:
  - Systems continue to make greater use of
    - Web resources: Wikipedia, answer repositories
    - Machine learning!!!!
Conclusions

- Deep processing for QA
  - Exploits parsing, semantics, anaphora, reasoning
  - Computationally expensive
    - But tractable because applied only to
    - Questions and Passages
Summary

- Deep processing techniques for NLP
  - Parsing, semantic analysis, logical forms, reference, etc
  - Create richer computational models of natural language
    - Closer to language understanding
Summary

- Deep processing techniques for NLP
  - Parsing, semantic analysis, logical forms, reference, etc
  - Create richer computational models of natural language
    - Closer to language understanding

- Shallow processing techniques have dominated many areas
  - IR, QA, MT, WSD, etc
    - More computationally tractable, fewer required resources
Summary

- Deep processing techniques for NLP
  - Parsing, semantic analysis, logical forms, reference, etc
  - Create richer computational models of natural language
    - Closer to language understanding

- Shallow processing techniques have dominated many areas
  - IR, QA, MT, WSD, etc
    - More computationally tractable, fewer required resources

- Deep processing techniques experiencing resurgence
  - Some big wins – e.g. QA
Summary

- Deep processing techniques for NLP
  - Parsing, semantic analysis, logical forms, reference, etc
  - Create richer computational models of natural language
    - Closer to language understanding

- Shallow processing techniques have dominated many areas
  - IR, QA, MT, WSD, etc
    - More computationally tractable, fewer required resources

- Deep processing techniques experiencing resurgence
  - Some big wins – e.g. QA
  - Improved resources: treebanks (syn/disc, Framenet, Propbank)
  - Improved learning algorithms: structured learners, ...
  - Increased computation: cloud resources, Grid, etc
Notes

- Last assignment posted – Due March 15
  - No coding required

- Course evaluation web page posted:
  - Please respond!
  - https://depts.washington.edu/oeaias/webq/survey.cgi?user=UWDL&survey=1254

- THANK YOU!