Strategies for QA & Information Retrieval

Ling573
NLP Systems and Applications
April 10, 2014
Roadmap

- Shallow and Deep processing for Q/A
  - AskMSR, ARANEA: Shallow processing Q/A
    - Wrap-up
  - PowerAnswer-2: Deep processing Q/A

- Information Retrieval:
  - Problem:
    - Matching Topics and Documents
  - Methods:
    - Vector Space Model
  - Retrieval evaluation
Redundancy-based Answer Extraction

- Prior processing:
  - Question formulation
  - Web search
  - Retrieve snippets – top 100

- N-grams:
  - Generation
  - Voting
  - Filtering
  - Combining
  - Scoring
  - Reranking
N-gram Filtering

- Throws out ‘blatant’ errors
  - Conservative or aggressive?
    - Conservative: can’t recover error

- Question-type-neutral filters:
  - Exclude if begin/end with stopword
  - Exclude if contain words from question, except
    - ‘Focus words’: e.g. units

- Question-type-specific filters:
  - ‘how far’, ‘how fast’: exclude if no numeric
  - ‘who’, ‘where’: exclude if not NE (first & last caps)
N-gram Filtering

- Closed-class filters:
  - Exclude if not members of an enumerable list
  - E.g. ‘what year ‘ -> must be acceptable date year

- Example after filtering:
  - Who was the first person to run a sub-four-minute mile?

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bannister</td>
<td>137</td>
</tr>
<tr>
<td>Roger</td>
<td>114</td>
</tr>
<tr>
<td>Roger Bannister</td>
<td>103</td>
</tr>
<tr>
<td>English</td>
<td>26</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
N-gram Combining

• Current scoring favors longer or shorter spans?
  • E.g. Roger or Bannister or Roger Bannister or Mr.....
    • Bannister pry highest – occurs everywhere R.B. +

• Generally, good answers longer (up to a point)

• Update score: $S_c += \sum S_t$, where $t$ is unigram in $c$

• Possible issues:
  • Bad units: Roger Bannister was – blocked by filters
    • Also, increments score so long bad spans lower

• Improves significantly
N-gram Scoring

- Not all terms created equal
  - Usually answers highly specific
  - Also disprefer non-units

- Solution: IDF-based scoring
  \[ S_c = S_c \times \text{average\_unigram\_idf} \]

<table>
<thead>
<tr>
<th>After combining</th>
<th>After scoring</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Candidate</strong></td>
<td><strong>Score</strong></td>
</tr>
<tr>
<td>Roger Bannister</td>
<td>354</td>
</tr>
<tr>
<td>Sir Roger Gilbert Bannister</td>
<td>286</td>
</tr>
<tr>
<td>Sir Roger Bannister</td>
<td>280</td>
</tr>
<tr>
<td>Bannister Sir Roger</td>
<td>278</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
N-gram Reranking

- Promote best answer candidates:
  - Filter any answers not in at least two snippets
  - Use answer type specific forms to raise matches
    - E.g. ‘where’ -> boosts ‘city, state’

- Small improvement depending on answer type
Summary

- Redundancy-based approaches
  - Leverage scale of web search
  - Take advantage of presence of ‘easy’ answers on web
  - Exploit statistical association of question/answer text

- Increasingly adopted:
  - Good performers independently for QA
  - Provide significant improvements in other systems
    - Esp. for answer filtering

- Does require some form of ‘answer projection’
  - Map web information to TREC document
Deliverable #2

• Baseline end-to-end Q/A system:
  • Redundancy-based with answer projection
    also viewed as
  • Retrieval with web-based boosting

• Implementation: Main components
  • (Suggested) Basic redundancy approach
  • Basic retrieval approach (IR next lecture)
Data

- Questions:
  - XML formatted questions and question series

- Answers:
  - Answer ‘patterns’ with evidence documents

- Training/Devtext/Evaltest:
  - Training: Thru 2005
  - Devtest: 2006
  - Held-out: ...

- Will be in /dropbox directory on patas

- Documents:
  - AQUAINT news corpus data with minimal markup
PowerAnswer2

- Language Computer Corp.
  - Lots of UT Dallas affiliates

- Tasks: factoid questions

- Major novel components:
  - Web-boosting of results
  - COGEX logic prover
  - Temporal event processing
  - Extended semantic chains

- Results: Best factoid system: 0.713 (vs 0.666, 0.329)
Challenges: Co-reference

- Single, basic referent:

- Multiple possible antecedents:
  - Depends on previous correct answers
Challenges: Events

- Event answers:
  - Not just nominal concepts
  - Nominal events:
    - Preakness 1998
  - Complex events:
    - Plane clips cable wires in Italian resort
- Establish question context, constraints
Handling Question Series

- Given target and series, how deal with reference?

- Shallowest approach:
  - Concatenation:
    - Add the ‘target’ to the question

- Shallow approach:
  - Replacement:
    - Replace all pronouns with target

- Least shallow approach:
  - Heuristic reference resolution
Question Series Results

- No clear winning strategy
  - All largely about the target
    - So no big win for anaphora resolution
    - If using bag-of-words features in search, works fine

- ‘Replacement’ strategy can be problematic
  - E.g. Target=Nirvana:
    - What is their biggest hit?
    - When was the band formed?
      - Wouldn’t replace ‘the band’

- Most teams concatenate
PowerAnswer-2

- Factoid QA system:
PowerAnswer-2

- Standard main components:
  - Question analysis, passage retrieval, answer processing
- Web-based answer boosting
- Complex components:
  - COGEX abductive prover
  - Word knowledge, semantics:
    - Extended WordNet, etc
  - Temporal processing
Web-Based Boosting

- Create search engine queries from question
- Extract most redundant answers from search
  - Cf. Dumais et al - AskMSR; Lin – ARANEA
- Increase weight on TREC candidates that match
  - Higher weight if higher frequency
- Intuition:
  - Common terms in search likely to be answer
  - QA answer search too focused on query terms
  - Reweighting improves
- Web-boosting improves significantly: 20%
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
Deep Processing:
Answer Selection

- Cogex prover:
  - Applies abductive inference
    - Chain of reasoning to justify the answer given the question
    - Mix of logical and lexical inference

- Main mechanism: 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.
  - Yields 12% improvement in accuracy!
Example

- How hot does the inside of an active volcano get?
- Get(TEMPERATURE, inside(active(volcano)))
- “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

Knowledge derived from WordNet to proof ‘axioms’

Ex. Due to D. Jurafsky
Temporal Processing

- 16% of factoid questions include time reference
- Index documents by date: absolute, relative
- Identify temporal relations b/t events
  - Store as triples of (S, E1, E2)
    - S is temporal relation signal – e.g. during, after
- Answer selection:
  - Prefer passages matching Question temporal constraint
  - Discover events related by temporal signals in Q & As
  - Perform temporal unification; boost good As
- Improves only by 2%
  - Mostly captured by surface forms
## Results

<table>
<thead>
<tr>
<th></th>
<th>PowerAnswer-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factoid</td>
<td>0.713</td>
</tr>
<tr>
<td>List</td>
<td>0.468</td>
</tr>
<tr>
<td>Other</td>
<td>0.228</td>
</tr>
<tr>
<td>Overall</td>
<td>0.534</td>
</tr>
</tbody>
</table>

Table 2: Results in the main task.