CENTRE FOR LANGUAGE TECHNOLOGY
MACQUARIE UNIVERSITY ~ SYDNEY

Answer me Exactly

Diego Mollá-Aliod

28 June 2004

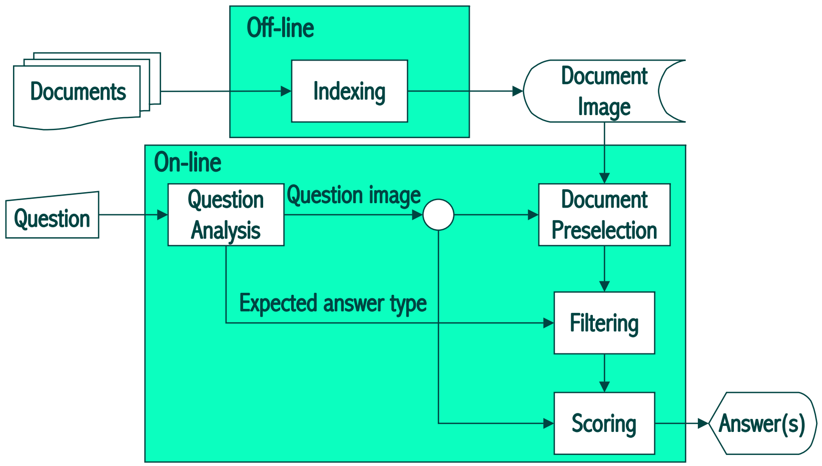
Outline

- Extracting Exact Answers
- Techniques
- The Proposal

Extracting Exact Answers

- Information Retrieval Tasks
  - Document Retrieval
  - Passage Retrieval
  - Extracting Passages Containing an Answer
  - Extracting the Exact Answer

AnswerFinder in TREC-QA 2003



Extracting Passages Containing an Answer

- Question Analysis
  - Determine the named entities of the expected answers
- Document Preselection
  - Use a third-party IR system
- Filtering
  - Preselect sentences containing a reasonable number of keywords
  - Reward sentences containing the right named entities
- Scoring
  - Sentence similarity measures

What is an Exact Answer?

- *What is the longest river in the United States?*
  - Correct and exact answers:
    - *Mississippi*
    - *the Mississippi*
    - *the Mississippi River*
    - *Mississippi River*
    - *mississippi*
  - Incorrect or inexact answers:
    - *At 2,348 miles the Mississippi River is the longest river in the US.*
    - *2,348 miles; Mississippi*
    - *the river Mississippi*
    - *Missipp*
    - *Missouri*

Outline

- Extracting Exact Answers
- Techniques
- The Proposal

The Easiest Technique

- Return string that matches your expected NE type
- Problems
  - What if there are no NEs of the correct type?
  - What if there are several NEs of the correct type?
  - Depends on the accuracy of the NE recogniser

The Winner of TREC-QA 2003

- Language Computer Corporation (LCC)'s QA system was (as usual) the best system in TREC-QA 2003
  - 70% accuracy for factoid questions
- Approach:
  1. Extract answers according to the built-in NE recogniser
  2. If the answer is not extracted as a NE, justify abductively
     - Use a logic tool

Using Logical Forms (Edinburgh University)

- Use of Discourse Representation Structures (DRSs)

```
x1008 x1003 e1004 x1006 e1009

rel(e1009, x1003, 'In')
answer(x1003, country)
pred(x1006, game)
rel(x1006, x1008, of)
pred(x1008, croquet)
arg1(e1009, x1006)
event(e1009, originate)
```

In
what
country
did
the
game
of
croquet
originate

- High scores for perfect matches
- Low scores if "relaxed" unification required
  - different semantic types
  - different argument order
  - symbols related according to WordNet
- Generate the answer by collecting the words pointed to by DRS conditions with discourse referents denoting the answer

A Combination of Methods (U. Southern California)

- Knowledge Based
- Pattern based
- Statistics-based

- Combined by maximum entropy based on 48 feature functions
  - Component-specific
  - Redundancy-specific
  - Qtarget-specific
  - Blatant-error-specific

Knowledge-based Method

- Determine the answer type or Qtarget
  - 185 different types organised in classes

Question: How tall is Mt. Everest?

Qtarget: DISTANCE-QUANTITY

Answer candidates:

- Jack knows exactly how tall Mt. Everest is
- Jack climbed the 29,028-foot Mt. Everest on 1984 and the 7,130-foot Mt. Kosciusko in Australia in 1985
- Mt. Everest is 2.8% taller than K2

Knowledge-based Method

- Reward answers with the same semantic relations

Question: Who killed Lee Harvey Oswald?
Text: Jack Ruby, who killed John F. Kennedy assassin Lee Harvey Oswald

- Use reformulation patterns

Question: How deep is Crater Lake?
Reformulation patterns:
  - Crater Lake is <what distance> deep?
  - depth of Crater Lake is <what distance>?
  - Crater Lake has a depth of <what distance>?
  - <what distance> deep Crater Lake?

Knowledge-based Method

- Possibility of reformulation chains

:anchor-pattern “SOMEBODY_1 is a student at COLLEGE_2”
:answers “Where does SOMEBODY_1 go to college?” :answer COLLEGE_2
:anchor-pattern “SOMEBODY_1 was a student at COLLEGE_2”
:can-be-inferred-from “SOMEBODY_1 dropped out of COLLEGE_2”
:anchor-pattern “SOMEBODY_1 dropped out of COLLEGE_2”
:is-equivalent-to “SOMEBODY_1 is a COLLEGE_2 dropout”

Text corpus: Bill Gates is a Harvard dropout
Original question: Where did Bill Gates go to college?
Reformulations:
  - Bill Gates was a student at <which college>
  - Bill gates dropped out of <which college>
  - Bill Gates is a <which college> dropout

Answer: Harvard

Pattern-based Method

- Idea: try to obtain as many patterns as possible
- Method: automatic learning of patterns

Given a Qtarget (a relation such as BIRTHYEAR), instantiated by a QA pair such as (NAME_OF_PERSON, BIRTHYEAR), extract from the web all the different patterns (TEMPLATEs) that contain this QA pair, and also determine the precision of each pattern

Extracting the Patterns

1. Submit a sample to a search engine
   - e.g. to learn the patterns for a pair (NAME_OF_PERSON BIRTHYEAR) submit “Gandhi 1869”
2. Retrieve the top 1,000 documents
3. Split the documents into sentences
4. Select only the sentences that contain both the question and the answer
5. Count phrases and sub-phrases
6. Retain phrases that contain both the question and the answer

Calculating the Precision of Each Pattern

1. The question term alone (without the answer) is given to a search engine
2. Retrieve the top 1,000 documents
3. Split the documents into sentences
4. Keep the sentences that contain the question terms
5. Select the sentences obtained in previous slide that match (pattern matching) the sentences obtained in step 4
6. Compute:

$$\text{Precision} = \frac{\text{\# patterns matching the answer (step 4 above)}}{\text{total \# patterns (see previous slide)}}$$

Integrating the Patterns into the QA System

- What happens if the user question has no Qtargets?
- Solution: use Maximum Entropy to score the patterns
  - Linear combination of the values of feature functions
- Features used:
  - Pattern
  - Frequency
  - Qtarget
  - Question word absent
  - Word match

$$P(a \mid \{a_1 a_2 .. a_A\}, q) = \frac{\exp[\sum_{m=1}^{M} \lambda_m f_m(a, \{a_1 a_2 .. a_A\}, q)]}{\sum_{a'} \exp[\sum_{m=1}^{M} \lambda_m f_m(a', \{a_1 a_2 .. a_A\}, q)]}$$

Statistics-based Method

- Use a noisy channel model
  - How can a given sentence S_A that contains the answer A be rewritten into the question Q?
- Like statistical Machine Translation
- But:
  - answer sentences are much longer than typical questions
  - answer sentences contain redundant information
- Approach:
  - Set a "cut" on the parse tree of the answer sentence

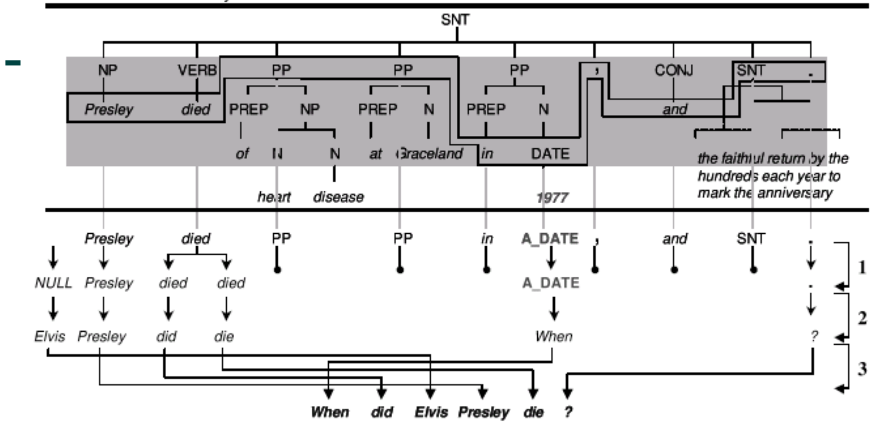
Statistical QA: Approach

1. Make a cut in the answer parse tree so that:
   - every word in the answer sentence or one of its ancestors belongs to the "cut", and
   - no two nodes on a path from a word to the root of the tree are in the "cut"
2. Mark one of the elements in the "cut" as the answer string
3. Match the elements of the cut with the question
   - assign fertility values to the cut elements
   - replace answer words with question words
   - permute the question words in order to obtain the grammatical question

What is the best cut?

- Terms overlapping with the question are preserved
- The answer is reduced to its semantic or syntactic class prefixed with "A_"
- Non-leaves that don't have any question term or answer offspring are reduced to their semantic or syntactic class
- All remaining nodes (leaves) are preserved as surface text

Q: *When did Elvis Presley die?*
S_A: Presley died of heart disease at Graceland in 1977, and the faithful return by the hundreds each year to mark the anniversary.



1) Fertility: n(1 | Presley) n(2 | died) n(0 | PP)…

2) Translation: t(Elvis | NULL) t(When | DATE)…

3) Distortion: d(1 | 5) d(2 | 3) d(3 | 1)…

Results of the Three Approaches

Metric	Knowledge-Based		Pattern-Based		Statistical-Based		Base from all systems followed by ME re-ranking (no feature selection)	Base from all systems followed by ME re-ranking (with feature selection)
	Base	Base followed by ME re-ranking	Base	Base followed by ME re-ranking	Base	Base followed by ME re-ranking		
Top answer	35.83%	45.03%	25.18%	30.50%	21.30%	32.20%	46.37%	47.21%
Top 5	57.38%	56.41%	35.59%	43.09%	31.23%	40.92%	57.62%	57.62%
MRR	43.88	49.36	28.57	35.37	24.83	35.51	51.07	51.27

Outline

- Extracting Exact Answers
- Techniques
- The Proposal

What do we Have in AnswerFinder?

- Question classifier
  - a *what*-question
  - expected answer: ??
- Named entity recogniser
  - *what do we have in PRODUCT-NAME?*
- Grammatical relations
  - (subj have we _) (aux inv have do) (ncmod answerfinder have in) (dobj have what _)
- Flat logical forms
  - holds(v_e4)~[]
    object('answerfinder',v_o6,[v_x6])~[6/'AnswerFinder']
    evt('have',v_e4,[v_x3,v_x1])~[1/'what',2/'do',…6/'AnswerFinder']
    object('what',v_o1,[v_x1])~[1/'what']
    object('we',v_o3,[v_x3])~[3/'we']
    prop('in',v_p5,[v_e4,v_x6])~[5/'in',6/'AnswerFinder']

The Approach

1. Use the standard process to preselect select 100 sentences
2. Select the answer candidates
   1. NE of the expected type
   2. patterns based on logical forms
3. Group similar answer candidates (substring match)
4. Combine the scores of the members of a group of answer candidates
   1. NE, answer patterns
   2. score of the host sentence

Logical Form Patterns

```
% generic “what”
answer_pattern_for_question_pattern(ANSWER,
         [object('what',_,[XWho])],
         [ [object(_,ANSWER,[XWho])] ]
).

% Q: When did International Volunteers Day begin?
% A: ``Cold Mountain'' began rising to the top of best-seller
% lists in 1997...
answer_pattern_for_question_pattern(ANSWER,
    [evt(EvtY,VeventY,ArgsY),
     prop(when,_VpropWHEN,[VeventY])],
    [
        [evt(EvtY,VeventY,ArgsY),
         prop(P,_VpropIN,[VeventY,VexistANSWER]),
         object(_Word,ANSWER,[VexistANSWER])],
        [evt(EvtY,VeventY,ArgsY),
         prop(P,_VpropIN,[VeventY,VexistANSWER]),
         dep(_ANSWER,ANSWER,[VexistANSWER])]
    ]
):-
       P = in ; P = on.
```


The Big Problems

- The NE recogniser does not mark the exact answer
  - Approaches to solve this:
    - return the most popular answer
    - try to normalise the NE output
    - use a customised NE recogniser
- Poor coverage of the patterns
  - Difficult to develop the patterns
  - Approaches to solve this:
    - simplify the patterns
    - develop a tool to facilitate discovery of patterns
    - automatic learning of patterns
      - Genetic approaches
      - Given a corpus of QA pairs