Question Answering

- Not the classic IR scenario
- Text Retrieval Conference TREC-8

“People have questions and they need answers, not only text. Automatic question answering will definitely be a significant advance in the state-of-art information retrieval technology.”

- Unstructured document corpus
- Answers as portions of surface text
- Ranking of top 5 answers
- 50- and 250-byte versions
Approaches used in TREC-8

Information Retrieval

- Application of existing IR technology to the new problem
- Query processed for keywords
  → president, USA
- Passages returned on statistical relevance measure $tf/idf$
- Some performance with 250-byte passages
- Poor performance with 50-byte passages
Approaches used in TREC-8

Query Processing & Named Entity Extraction

- Query processed for keywords and required entity class
  Who → PERSON
  When → TIME
  How long → DURATION, LENGTH

- Passages returned using IR techniques

- Entities within passages identified and assigned to semantic category
  Bill Clinton → PERSON

- Various heuristics for choice of entity

- Most popular and best performing approach
Problems revealed by TREC-8

Q 1 Who is the president of the USA?

A 1.1 Bill Clinton is the president of the USA.

A 1.2 Even if one disapproves of Bill Clinton, it is impossible to deny that although many people do not agree with him, he is one of the most powerful men on earth: he is the leader of the world and the president of the USA.

A 1.3 Hillary Clinton is the wife of the president of the USA.
Approaches used in TREC-8

Sentence Structure

- University of Maryland: dependency parser
  Dependency tree helps narrow entity choice
  Not enough to prevent selection of non-answers
- CL Research: semantic triples
  Semantic roles within sentence
  Restricted scope
  Full parser → not robust
  No coreference resolution
Aims of Project

- Use sentence structure to select answer entities and prepositions.
- Requirements:
  Robust (→ no full parse)
  Select portions of surface text as answer
- Simplifications:
  Ignored tense, inference, negation
- Assumptions:
  IR query assumed (only passages containing keyword)
  NEE assumed (entities added to lexicon)
System Overview

1. Query
2. Answer Passage
3. Shallow Text Processing
4. Structural Relation Extraction
5. Structural Matching
6. Decision/Answer
Shallow Text Processing

![Diagram of Shallow Text Processing]

Text

PoS Tagging

NP Grouping

External Module

Rewriting

Stemming

NP Adjustment

Front End

Interface

To Syntactic Processor

Figure 1: Front End
Figure 2: Main Functions
Syntactic Processing

Front End Tagging, stemming:

\[
\text{[np: [bill/mn], np: [clinton/np], be/vbz, np: [the of/prep, np: [the/det, usa/np], ‘.’ ‘.’ ‘.’]}
\]

VG/NG/PP Grouping Compound NPs, conjunctions, aux inversion

```
S
  /\  \\
NG  VG  NG
 /\  /\  /\ \\
NP NP V NP
```

```
Bill Clinton be the president P NP
```

```
of the USA
```
Semantic Processing

VG **Info** Predicate name, voice

NG/PP **Info** Semantic class, gender, number

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>per/f</td>
<td>Person (female)</td>
<td>queen, Mary</td>
</tr>
<tr>
<td>per/ e</td>
<td>Person (either)</td>
<td>student, candidate</td>
</tr>
<tr>
<td>obj</td>
<td>Concrete object</td>
<td>banana, kitchen</td>
</tr>
<tr>
<td>abs</td>
<td>Abstract object</td>
<td>election, answer</td>
</tr>
<tr>
<td>loc</td>
<td>Location</td>
<td>Wales, Cambridge</td>
</tr>
<tr>
<td>org</td>
<td>Organisation</td>
<td>university, Microsoft</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>loc</td>
<td>Location</td>
<td>in, outside</td>
</tr>
<tr>
<td>tim</td>
<td>Time</td>
<td>in, before</td>
</tr>
<tr>
<td>man</td>
<td>Manner</td>
<td>by</td>
</tr>
<tr>
<td>pos</td>
<td>Possession</td>
<td>in, of</td>
</tr>
<tr>
<td>xxx</td>
<td>Unknown</td>
<td>(anything)</td>
</tr>
</tbody>
</table>

**PP Attachment**

```
S
  /  
/    
NG: s/[per/m]  NG: s/[per/m]  VG: act#be  NG: s/[per/e]
  
NP: s/[per/m]  NP: s/[per/m]
  
Bill  Clinton

V

be

NP: s/[per/e]

P

PP: [loc, pos]

P

NP: s/ [per/ e]

of

the President

of

the University
```
Coreference Resolution

- Entity indexing - provide “pointers”

- Resolution of pronouns → NGs - by semantic class & number
- Resolution of proper/definite NGs & number anaphora — needing more info
- Resolution of time & location anaphora
- Non-deterministic
Syntactic Simplification

• Complex sentences split into list of simple “equivalents”

• Conjunctions, punctuation
  “John likes Mary but Mary likes Bill”
  → “John likes Mary”, “Mary likes Bill”

• Subordinate, relative clauses
  “Mary, although John likes her, likes Bill”
  → “Although John likes Mary”, “Mary likes Bill”
  “Mary, who John likes, likes Bill”
  → “John likes Mary”, “Mary likes Bill”

• Queries & answers require different treatment later
Structure Extraction

- Predicate-argument structures formed from NG-VG-NG
  
  \[
  s:\ [ \text{like, john, mary} ]
  \]
  
  \[
  s:\ [ \text{like, who, mary} ]
  \]

- State structures formed if VG is existential, or from NG
  
  \[
  s:\ [ \text{[bill clinton]}, [[\text{president}], [of, [the ]}]
  \]
  
  \[
  s:\ [ \text{who, [[\text{president}], [of, [the usa]]]} ]
  \]
Lexical Matching

- Query words matched with suitable entities by semantic type:

<table>
<thead>
<tr>
<th>where</th>
<th>loc</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>when</td>
<td>tim</td>
<td>PP, NG, NP</td>
</tr>
<tr>
<td>who, whom</td>
<td>per, org</td>
<td>NG, NP</td>
</tr>
<tr>
<td>what</td>
<td>abs, obj, org</td>
<td>NG, NP</td>
</tr>
<tr>
<td>which</td>
<td>abs, obj, org</td>
<td>NG, NP</td>
</tr>
<tr>
<td>whose</td>
<td>pos</td>
<td>PP</td>
</tr>
<tr>
<td>how</td>
<td>man</td>
<td>PP, AdvP</td>
</tr>
<tr>
<td>why</td>
<td>rea</td>
<td>PP</td>
</tr>
</tbody>
</table>

- VGs matched by predicate name
- NGs matched by content (complex NGs require many rules, etc.)
- PPs matched by semantic class and NG matching (including stacking)
Structural Matching

- Set of equivalence rules for tight matching of significantly different structures
- Active-Passive
  \[ s : [VG_{pas}, Arg1, pp : [by, Arg2]] ⇔ s : [VG_{act}, Arg1, Arg2] \]
- Existential Ordering
  \[ s : [Arg1, Arg2] ⇔ s : [Arg2, Arg1] \]
- Verb Nominalisation
  \[ s : [VG_{util}, Arg1, Arg2, ...] ⇔ s : [VG_{lemma}, Arg1', ...] \]
  where \( Arg1' = Arg1 \setminus NP_{lemma} \)
- PP-Verbs (e.g. possession)
  \[ s : [VG_{pos}, Arg1, Arg2] ⇔ s : [Arg2, pp : [pos] : [o]] \]
## Answer Passage Phenomena

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Match</td>
<td>“Snowdon is in Wales”</td>
</tr>
<tr>
<td>Singular/Plural</td>
<td>“There are several racecourse in Newmarket”</td>
</tr>
<tr>
<td>Unnecessary Modifiers</td>
<td>“Snowdon, the highest mountain in the UK, is in Wales”</td>
</tr>
<tr>
<td>Stem Matching</td>
<td>“The biggest IT company is Microsoft”</td>
</tr>
<tr>
<td>Inclusions</td>
<td>“The Pacific islands like Hawaii produce tropical fruit like bananas”</td>
</tr>
<tr>
<td>Expansions</td>
<td>“In the Suffolk town of Newmarket there is a racecourse”</td>
</tr>
<tr>
<td>Simple PPs</td>
<td>“Livingstone is mayor of London”</td>
</tr>
<tr>
<td>Possessive PPs</td>
<td>“Ken Livingstone defeated all the other candidates and is Lord…”</td>
</tr>
<tr>
<td>Verb Form Variations</td>
<td>“Hawaii is producing bananas”</td>
</tr>
<tr>
<td>VG Modifiers</td>
<td>“Hawaii produced bananas in 1991”</td>
</tr>
<tr>
<td>Verbs of Possession</td>
<td>“Newmarket has a racecourse”</td>
</tr>
<tr>
<td>Passives</td>
<td>“One is made very fat by bananas”</td>
</tr>
<tr>
<td>Existential Ordering</td>
<td>“In Newmarket there is a racecourse”</td>
</tr>
<tr>
<td>Existential Compounds</td>
<td>“Snowdon, in Wales, is a serious mountain”</td>
</tr>
<tr>
<td>Coreference</td>
<td>“Pat is in the kitchen and Mike is there too”</td>
</tr>
<tr>
<td>Conjunctions</td>
<td>“Pat is in the lounge and Mike is in the garden”</td>
</tr>
<tr>
<td>Relative Clauses (internal)</td>
<td>“Hawaii specializes in the production of tropical fruit, which includes bananas”</td>
</tr>
<tr>
<td>Relative Clauses (external)</td>
<td>“Pat, who is reading, is in the lounge”</td>
</tr>
<tr>
<td>Relative Clauses (coindexed)</td>
<td>“Tropical fruit production, which includes banana production in Hawai”</td>
</tr>
<tr>
<td>Subordinate Clauses</td>
<td>“There is a library, with all the texts the student needs, in Cambridge”</td>
</tr>
<tr>
<td>Verb Nominalisation</td>
<td>“In Hawaii, which is not far from California, there is large scale bananas”</td>
</tr>
</tbody>
</table>
Evaluation

- Answers were assessed by a “notional user” with no detailed knowledge of the system
- Portions of surface text marked manually and compared to output
- Difficulty of consistency in marking → “narrow-scope” overlap allowed
- Conventional IR performance measures calculated:
  
  \[
  \text{Recall} = \frac{n(\text{correct} \& \text{identified})}{n(\text{correct})}
  \]

  \[
  \text{Precision} = \frac{n(\text{correct} \& \text{identified})}{n(\text{identified})}
  \]

- 86% recall, 97% precision on training data
- 50-80% recall, >90% precision in blind test (cf. TREC-8, 5 answers)
Summary

- Structural relations used successfully, especially to reject
- Structural equivalences useful in the majority of cases
- Requires complex coding of rules, so large amounts of training
  robust to unseen structural phenomena
  “Who is the author of the book [...]?” ↔ “[...] by [...]
- What do we do about inference & negation?
  “about 7 inches higher than 14,776 feet 2 inches” ↔ “inches”