Question Answering: What is the Best Scoring Mechanism?

Diego Mollá-Aliod
17 Feb. 2003

Outline

• Architecture of our QA system
• Scoring modules
• The scoring tyranny
• The effect of linguistic information
• Discussion

Architecture of a QA System

The Document Set

• Remedia Publications’ Reading Comprehension Tests
• Levels 2, 3, 4, and 5
• Every document contains 5 questions (who, what, when, where, why)
• The answer is always in the text
• Files:
  – Original text (*.txt)
  – Coreferences (*.txt.coref)
  – Named Entities (*.txt.ne)
  – Answers marked-up (*.txt.snr)
  – Answers extracted (*.txt.wdra.xml)
Watch Out for Sears!
(North Redwood, Minn., September, 1889) A man named Richard Sears has been playing a joke on shoppers. Sears likes to sell items by mail. Not long ago, he ran an ad in some newspapers in small towns. The ad showed a drawing of lovely furniture. There was a sofa and two chairs.

The ad said the furniture was for sale. It said the pieces were made of fine metal frames and were beautiful to see. The ad said that for a short time only, these chairs would be shipped to all who paid 95 cents.

This message sounded too good to be true. Still, a lot of people sent in their money. Imagine their surprise when they received the furniture. The furniture was made for a dollhouse! They were tiny pieces.

Some people complained. That's when Sears showed them the tiny print in his ad. In very small letters, he had included the word "miniature." That means the furniture was not full size. Sears says he did this to get attention.

1. Who played a joke on shoppers?
2. What does the Sears ad offer?
3. When did Sears play this joke?
4. Where is the word "miniature"?
5. Why did Sears play this joke?
Our QA Prototype

def answer_extraction(question_data, document_data):
    "Perform answer extraction"
    (question, expected_answer, question_analysed) = question_data
    scores = []
    for (sentence, sentence_analysed) in document_data:
        if ne_filtering(sentence, expected_answer):
            score = -100
        else:
            score = 0
        (sc, sc_justification) = call_scoring_method(question_analysed, sentence_analysed)
        score += sc
        scores += [(score, sc_justification, sentence)]
    scores.sort(lambda x,y: cmp(y[0], x[0]))
    return scores[0:5]

Outline

- Architecture of our QA system
- Scoring modules
  - The scoring tyranny
  - The effect of linguistic information
  - Discussion

Word, Dependency Overlap

- Word overlap
  - With/without stop words
  - With/without stemming
- Dependency overlap
  - A man named Richard Sears has been playing a joke on shoppers.
    man>play, name>man, richard>sears, sears>name, joke>play, on>play, shopper>on
  - Who played a joke on shoppers?
    who>play, joke>play, on>play, shopper>on

Finding the Dependencies

- Output of Conexor
  1 A a det:>2 @DN> %>N DET SG
  2 man man subj:>5 @SUBJ %NH N NOM SG
  3 named name mod:>2 @-FMAINV %VP EN
  4 Richard richard obj:>3 @OBI %NH N NOM SG
  5 Sears sear subj:>6 @+FMAINV %VA V PRES SG3
  6 has have v-ch:>7 @-FAUXV %AUX V PRES SG3
  7 been be v-ch:>8 @-FAUXV %AUX EN
  8 playing play main:>0 @-FMAINV %VA ING
  9 a a det:>10 @DN> %>N DET SG
  10 joke joke comp:>8 @PCOMPL-S %NH N NOM SG
  11 on on phr:>8 @ADVL %EH PREP
  12 shoppers shopper pcomp:>11 @<P %NH N NOM PL
  13 <s> <s>
Finding the Dependencies

- Dependencies:
  - a>man, man>sear, name>man, richard>name,
    sear>have, have>be, be>play, a>joke, joke>play,
    on>play, shopper>on
- Dependencies removing stop words (a, have, be):
  - man>sear, name>man, richard>name, sear>play,
    joke>play, on>play, shopper>on

Grammatical Relations (Carroll et al. 1998)

Grammatical Relations

- A man named Richard Sears has been playing a joke on shoppers.
  (xmod _ man name), (detmod _ man a),
  (subj name man), (dobj name richard _),
  (detmod _ joke a), (subj sear man _), (subj play sear _),
  (aux _ play have), (aux _ play be), (ncmod _ play on),
  (xcomp _ play joke)

- Who played a joke on shoppers?
  (subj play who _), (dobj play joke _), (ncmod _ play on),
  (detmod _ joke a)

Minimal Logical Forms

- A man named Richard Sears has been playing a joke on shoppers.
  holds(v_o10), object('man',v_o2,[v_x2]),
  evt('name',v_e3,[v_X3,v_x4,v_x2]),
  object('joke_on',v_o10,[v_e5,v_x12]),
  object('richard',v_o4,[v_x4]),
  object('shopper',v_o12,[v_x12])

- Who played a joke on shoppers?
  holds(v_e2), object('who',v_o1,[v_x1]),
  evt('play_on',v_e2,[v_x1,v_x4,v_x6]),
  object('joke',v_o4,[v_x4]),
  object('shopper',v_o6,[v_x6])
Parser Accuracy

- Corpus:
  - 500 sentences
  - 10,000 words
- Used in Carroll & Briscoe's parser evaluation

First Results

<table>
<thead>
<tr>
<th>Overlap</th>
<th>Depend.</th>
<th>Gramrels</th>
<th>MLFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 3</td>
<td>Level 4</td>
<td>Level 5</td>
<td></td>
</tr>
</tbody>
</table>

Outline

- Architecture of our QA system
- Scoring modules
  - The scoring tyranny
- The effect of linguistic information
- Discussion

Best and Worst Cases

- The overlap scores in the gramrels and MLFs are very low
- The right answer is competing with other sentences

<Q>What does the Sears ad offer?</Q>

<ANS1 score="1" score_justification="[compound_noun(v_x2,v_x3)]" correct="no">1989 Remedia Publications, Comprehension/5W<92>s-4</ANS1>

<ANS2 score="1" score_justification="[compound_noun(v_x2,v_x3)]" correct="no">(North Redwood, Minn</ANS2>

<ANS3 score="1" score_justification="[object(ad,v_o7,[v_x7])]" correct="no">Not long ago, he ran an ad in some newspapers in small towns</ANS3>

<ANS4 score="1" score_justification="[object(ad,v_o2,[v_x2])]" correct="no">The ad showed a drawing of lovely furniture</ANS4>

<ANS5 score="1" score_justification="[object(ad,v_o2,[v_x2])]" correct="yes">The ad said the furniture was for sale</ANS5>
Combination of Methodologies

```python
def combo_overlap(q_data, s_data):
    "Combination of overlap scores"
    v_overlap = overlap(q_data[0], s_data[0])[0]
    v_dep_overlap = dep_overlap(q_data[1], s_data[1])[0]
    v_mlf_overlap = mlf_overlap(q_data[2], s_data[2])[0]
    v_gr_overlap = gramrel_overlap(q_data[3], s_data[3])[0]
    return v_overlap + v_dep_overlap*3 + v_mlf_overlap*9 + v_gr_overlap*27,''

def combo2_overlap(q_data, s_data):
    "Combination of overlap scores"
    v_overlap = overlap(q_data[0], s_data[0])[0]
    v_dep_overlap = dep_overlap(q_data[1], s_data[1])[0]
    v_mlf_overlap = mlf_overlap(q_data[2], s_data[2])[0]
    v_gr_overlap = gramrel_overlap(q_data[3], s_data[3])[0]
    return v_overlap + v_dep_overlap*3 + v_gr_overlap*9 + v_mlf_overlap*27,''
```

With Combined Methods

```
Overlap Depend. GRs MLFs GR-C MLF-C
```

Detecting Best and Worst Cases

- The overlap scores in the grammars and MLFs are very low
- The right answer is competing with other sentences

```latex
<Q2>
\textit{What does the Sears ad offer?}
<Q>
\textit{ANS1 score=1} score_justification=[compound_noun(v_x2,v_x3)] correct="no">1989 Remedia Publications, Comprehension/5W<92>/s</ANS1>
\textit{ANS2 score=1} score_justification=[compound_noun(v_x2,v_x3)] correct="ns">(North Redwood, Minn</ANS2>
\textit{ANS3 score=1} score_justification=[object(ad_v_o7,[v_x7])] correct="no">Not long ago, he ran an ad in some newspapers In small towns</ANS3>
\textit{ANS4 score=1} score_justification=[object(ad_v_o7,[v_x7])] correct="no">The ad showed a drawing of lovely furniture</ANS4>
\textit{ANS5 score=1} score_justification=[object(ad_v_o7,[v_x7])] correct="yes">The ad said the furniture was for sale</ANS5>
<Q2>
```

New Results (average of all levels)

```
Overlap Depend. GRs MLFs GR-C MLF-C
```

- Given
Outline

- Architecture of our QA system
- Scoring modules
- The scoring tyranny
  - The effect of linguistic information
- Discussion

The Effect of Stemming

<table>
<thead>
<tr>
<th>Overlap (base forms)</th>
<th>Overlap (Porter stemmer)</th>
<th>Overlap (word forms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0.05</td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td>0.1</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>0.15</td>
<td></td>
<td>0.15</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td>0.25</td>
</tr>
<tr>
<td>0.3</td>
<td></td>
<td>0.3</td>
</tr>
<tr>
<td>0.35</td>
<td></td>
<td>0.35</td>
</tr>
<tr>
<td>0.4</td>
<td></td>
<td>0.4</td>
</tr>
<tr>
<td>0.45</td>
<td></td>
<td>0.45</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>0.5</td>
</tr>
</tbody>
</table>

Stop Words or No Stop Words in the Dependencies?

<table>
<thead>
<tr>
<th>Removing Stop Words</th>
<th>Untreated</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>0.45</td>
<td></td>
</tr>
</tbody>
</table>

What is the impact of the semantic dependences?
Why the negative impact of the semantic dependencies??

• Differences of scoring between the MLF and the MLF-base overlaps
  
  – Q: When did Sears play this joke?
  prop('when',v_p1,[v_e4]), evt('play',v_e4,[v_x3,v_x6]), object('sear',v_o3,[v_x3]), object('joke',v_o6,[v_x6])
  A: Watch Out for Sears
  evt('watch_out',v_e1,[v_X1]), evt('sear',v_e4,[v_X1])
  MLF overlap: []
  MLF base overlap: sear

• MLFs scored down wrongly: 5 + 9 +
• MLFs scored down rightly: 0 + 2 +

Q: Who had a dream to build a family park?
object('who',v_o1,[v_x1]), evt('have',v_e2,[v_x1,v_x4]), object('dream',v_o4,[v_x4]), event('build',v_e6,[v_x4,v_x9]), object('family',v_o8,[v_x8]), object('park',v_o9,[v_x9]), compound_noun(v_x8,v_x9)
A: He wanted to build a place where the whole family could have fun together
olds(v_e2), object('he',v_o1,[v_x1]), evt('want',v_e2,[v_x1,v_e4]), object('family',v_o10,[v_x10]), event('build',v_e4,[v_x1,v_x6]), event('have',v_e12,[v_x10,v_x13,v_x6]), object('place',v_o6,[v_x6]), object('fun',v_o13,[v_x13]), prop('where',v_p7,[v_e12]), prop('together',v_p14,[v_e12]), prop('whole',v_p9,[v_x10])
MLF overlap: [evt(build,v_e4,[v_x1,v_x6]),object(family,v_o10,[v_x10])]
MLF base overlap: build have family

Outline

• Architecture of our QA system
• Scoring modules
• The scoring tyranny
• The effect of linguistic information
• Discussion

Discussion

• One document per question, is that good for the evaluation?
• What is the impact of errors in parsing and semantic interpretation?
• Are the questions too simple?
• Is the text too simple?