NTT’s Question Answering System for NTCIR-6 QAC-4

Ryuichiro Higashinaka
Hideki Isozaki
NTT Communication Science Labs.
NTT Corporation
Overview

• Our two systems:
  – NCQAW-1 and NCQAW-2
    (NTT CS Labs’ QA System for ‘Why’ Questions)

• Features
  – Focus on ‘why’ questions
  – Machine learning approach for ‘why’ and ‘how’ questions
  – Pattern-based approach for ‘definition’ questions

• Achieve good performance on ‘why’ and ‘definition’ questions
Systems

• NCQAW-1
  – ML-based approach for ‘why’ and ‘how’
  – Pattern-based approach for ‘definition’
  – Uses SAIQA-QAC2 (our factoid QA system) for other question types
  – Question type analysis is based on rules

• NCQAW-2
  – Same as NCQAW-1 except that ‘why’ and ‘how’ questions are handled by rules
### Results

**NCQAW-1**

<table>
<thead>
<tr>
<th>question type</th>
<th>all</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>no output</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>24</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>other</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>why</td>
<td>38</td>
<td>11</td>
<td>6</td>
<td>0</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>how</td>
<td>26</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>total</td>
<td>100</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>38</td>
<td>32</td>
</tr>
</tbody>
</table>

Could not answer many ‘how’ questions because of question analysis failures.

**NCQAW-2**

<table>
<thead>
<tr>
<th>question type</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>no output</th>
</tr>
</thead>
<tbody>
<tr>
<td>why</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>how</td>
<td>26</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>total</td>
<td>16</td>
<td>3</td>
<td>2</td>
<td>46</td>
<td>33</td>
</tr>
</tbody>
</table>

Improvement by ML-based approach.
‘Why’ questions

• There are few systems for answering open-domain ‘why’ questions

• Previous approach
  – Extract causal sentences by hand-crafted rules as answer candidates (e.g., using cue words such as ‘tame’, ‘node’ etc.)
  – Rank the candidates by their similarity to the question

• Systems based on the approach
  – System by Morooka and Fukumoto (2006)
  – NCQAW-2
Problem

- Hand-crafted rules are costly to make
  - Cue words are not always reliable
    - Only 6-7 % of words before ‘de (by)’ are causes
      (Abekawa and Okumura, 2004)
    - Difficult to cover all causal expressions by hand

- Difficult to express degree of causality
  - Some expressions are more strongly expressing causality than others
    - ‘no riyuude (by reason of) ’
      vs. ‘kara (from)’, ‘tame (for)’, ‘de (by)’, etc.
Approach

- Adopt a machine learning approach to learn a causal sentence classifier

Diagram:
- Document Retrieval
  - Question
  - Top-N documents
- Answer Candidate Extraction
  - All sentences as answer candidates
- Causal Sentence Classifier
  - Causality score of the candidate
  - Similarity score between the question and the candidate
- Answer Evaluation
  - Top-N answer candidates

Corpus

Training

Outputs a causality score encoding how likely a cause is expressed within a sentence

(NB. This is a sentence extraction task.)
Formulation

(1) $\text{causal}_{\text{why}}(C)$

Causality score output by the causal sentence classifier

(2) $\text{sim}_{\text{why}}(S) = \sum_{w \in Q(S)} \text{idf}(w)$

Similarity score

$\text{sim}'_{\text{why}}(C) = 1/(1 + \exp(-\text{sim}_{\text{why}}(C)))$

N.B., Similarity score is normalized by the sigmoid function

Final score of an answer candidate:

$c\text{and score}_{\text{why}}(C') = \text{causal}_{\text{why}}(C) + \text{sim}'_{\text{why}}(C)$
Causal Sentence Classifier

• Use EDR Japanese corpus for training
  – Has annotation of ‘cause’ relation
  – 8,064 sentences with ‘cause’ out of 0.2M total sents.
  – Sentences with ‘cause’ → positive examples
    without ‘cause’ → negative examples

• Train a classifier by BACT
  – In sentence classification tasks, lexical, syntactic, and semantic features are useful
  → Adopt tree feature representation of a sentence
  → BACT: a boosting algorithm for classifying trees
    (uses existence of sub-trees as weak learners)
Tree feature representation of a sentence

Sentence:
彼は詐欺で逮捕された。
(He was arrested for fraud.)

Word Node:
- re
- ta
- EOS

Named-entity Node:
- Kare
- wa

Dependency relation:
- de
- Taiho
- Sa

POS Node:
- POS FuncW
- POS Verb

Semantic category Node:
- Arrest
- Execution

POS tag Node:
- Pronoun
- Verb
- Noun
- S-noun
Evaluation of the classifier

<table>
<thead>
<tr>
<th></th>
<th>rule-based</th>
<th>BACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>92.1%</td>
<td>96.0%</td>
</tr>
<tr>
<td>Precision</td>
<td>11.0%</td>
<td><strong>41.0%</strong></td>
</tr>
<tr>
<td>Recall</td>
<td>14.7%</td>
<td>4.2%</td>
</tr>
<tr>
<td>F-measure</td>
<td>12.6%</td>
<td>7.6%</td>
</tr>
</tbody>
</table>

(Rule-based: our implementation of Morooka and Fukumoto’s causal sentence extraction rules)

- Higher precision achieved by BACT
- Precision may be important from NCQAW-1’s results
- F can be raised to ca. 25% by feature engineering
Acquired Salient Patterns

- Acquired 178 patterns
- Cover most of expressions easily conceivable by humans
- Plus, include many other complex expressions
- Automatically disambiguating cue words e.g., ‘de’ (see below)

<table>
<thead>
<tr>
<th>rank</th>
<th>String-encoded subtree</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EOS 。 が</td>
<td>0.032</td>
</tr>
<tr>
<td>2</td>
<td>で 疑い (by suspicion of)</td>
<td>0.022</td>
</tr>
<tr>
<td>3</td>
<td>によって (by)</td>
<td>0.008</td>
</tr>
<tr>
<td>4</td>
<td>EOS 。 が 助詞-格助詞-一般</td>
<td>0.007</td>
</tr>
<tr>
<td>5</td>
<td>ため (for)</td>
<td>0.007</td>
</tr>
<tr>
<td>6</td>
<td>による (because of)</td>
<td>0.006</td>
</tr>
<tr>
<td>7</td>
<td>により (because of)</td>
<td>0.006</td>
</tr>
<tr>
<td>8</td>
<td>ので (because of)</td>
<td>0.006</td>
</tr>
<tr>
<td>9</td>
<td>から (from, because of)</td>
<td>0.005</td>
</tr>
<tr>
<td>10</td>
<td>。</td>
<td>0.005</td>
</tr>
<tr>
<td>11</td>
<td>動詞-自立 (verb)</td>
<td>0.004</td>
</tr>
<tr>
<td>12</td>
<td>2419 [types of illness]</td>
<td>0.004</td>
</tr>
<tr>
<td>13</td>
<td>。 。 の (of)</td>
<td>0.003</td>
</tr>
<tr>
<td>14</td>
<td>。 で 助動詞 (by)</td>
<td>0.003</td>
</tr>
<tr>
<td>15</td>
<td>から こと (from the fact that …)</td>
<td>0.003</td>
</tr>
</tbody>
</table>

\( \alpha \) : Weights given to each pattern
Examples of Answers

• NCQAW-2 extracts sentences with a strong cue
  – QAC4-00030: What is the purpose of the green power marketing?
  – A: さらに、原子力や石油への依存度を少しでも下げるため、電源の多様化を図るのは時代の流れだろう。

• NCQAW-1 can find answers without such cues
  – QAC4-00026: Why is the movable weir needed in the Yoshino river?
  – A: 可動堰化計画は、第十堰の老朽化が進んで洪水時の障害になり、壊れる危険もある、などとして持ち上がった。
‘How’ questions

• Same process as ‘why’ questions

• ‘condition’ relation is used instead of ‘cause’ to train a sentence classifier
  – sentences having ‘condition’ may have answers within the sentence

• Q: 聖火が消えたらどうしますか？
  (How do we cope when the Olympic flame burns out?)

• A: 聖火が消えたら再点火します.
  (If the Olympic flame burns out, it is reignited.)

Due to question type analysis failures, most of ‘how’ questions were not answered at all
‘Definition’ questions

• There are many systems for answering ‘definition’ questions (e.g., in TREC)

• Common approach
  – Extract descriptive sentences/nuggets as answer candidates using patterns or ML
  – Rank the candidates using importance of keywords within the candidate (i.e., the more keywords, the better)

• We adopt a similar approach
Answer candidate extraction

- Use patterns to find descriptive phrases
- Perform dependency tree matching to obtain phrases with all their modifiers

**Pattern Generation**

- Word(s) to define
  - Daisuke Matsuzaka
e.g., 松坂大輔

**Pattern Matcher**

- All matched phrases as answer candidates

---

**Mainichi news articles**

- 順  松坂大輔
  - [躍る春] 第70回記念センバツ 選手編 松坂大輔投手
  - P L学園（大阪）・稲田学投手、横浜（神奈川）・松坂大輔投手
  - 最速150キロの速球にプロも注目する横浜（神奈川）のエース・松坂大輔君
  ...

**Dependency tree matching**

- Tgrep2

**Generated patterns**

- 松坂大輔
- 「X」（松坂大輔）
- 松坂大輔（X）
- 松坂大輔というX
- 松坂大輔といったX
- 松坂大輔以外のX
- 松坂大輔などのX
- 松坂大輔等のX
- 松坂大輔のようなX
- 松坂大輔の様なX
- 松坂大輔といえばX
- 松坂大輔と言えばX
Answer Evaluation

• Answer candidates are ranked based on importance of words within the candidates

• Importance of words:

\[
\text{wordscore}_{\text{def}}(w) = \log(\text{tf}(w; \{C_i\}))
\]

(Term frequency within all answer candidates)

• Score of an answer candidate:

\[
\text{candscore}_{\text{def}}(C) = \sum_{w \in \text{CW}(C)} \text{wordscore}_{\text{def}}(w)
\]

(Sum of wordscores within the answer candidate)
Examples of answers

• QAC4-00018: スケルトンとはどのような競技ですか。
  (What is Skelton, the competitive sport?)
  – A: 54年ぶりに冬季五輪の正式種目として復活する、
    そり競技のスケルトン。
  – A: うつぶせでソリに乗ってボブスレーやリュージュと同じ
    コースを滑走する氷上競技「スケルトン」

• QAC4-00034: 「スポット傍受」とはどういったものですか。
  (What is the spot wiretapping?)
  – A: 通話内容が犯罪と関係あるか試し聴きする「スポット傍受」
  – A: 会話の冒頭（ぼうとう）を試験的に傍受する「スポット傍受」

• Adnominal/adverbial phrases by including modifiers
• Achieving good performance (9 A-rated answers/24)
## Results

**• NCQAW-1**

<table>
<thead>
<tr>
<th>question type</th>
<th>all</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>no output</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>24</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>other</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>why</td>
<td>38</td>
<td>11</td>
<td>6</td>
<td>0</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>how</td>
<td>26</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>total</td>
<td>100</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>38</td>
<td>32</td>
</tr>
</tbody>
</table>

Improvement by ML-based approach

Good performance by finding adnominal/adverbial phrases by dep. tree matching

Could not answer many ‘how’ questions because of question analysis failures

<table>
<thead>
<tr>
<th>question type</th>
<th>all</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>no output</th>
</tr>
</thead>
<tbody>
<tr>
<td>why</td>
<td>38</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>how</td>
<td>26</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>total</td>
<td>100</td>
<td>16</td>
<td>3</td>
<td>2</td>
<td>46</td>
<td>33</td>
</tr>
</tbody>
</table>
Conclusion

• Our two systems:
  – NCQAW-1 and NCQAW-2
• Features
  – Focus on ‘why’ questions
  – Machine learning approach for ‘why’ and ‘how’ questions
  – Pattern-based approach for ‘definition’ questions
• Achieved good performance on ‘why’ and ‘definition’ questions