A method for GeoTime information retrieval based on question decomposition and question answering

--- Yokohama National University at NTCIR-8 GeoTime ---

Tatsunori Mori
Yokohama National University, Japan
mori@forest.eis.ynu.ac.jp
Introduction

• We participated in the Japanese mono-lingual (JA-JA) task.

• Our proposed method for GeoTime information retrieval is based on
  – Question decomposition and
  – Question answering.

  • GeoTime information retrieval can be regarded as one special case of IR4QA, because a query submitted to a system is a natural language question in typical situations.
  • We may straightforwardly consider documents that have good answer candidates as documents relevant to the query.

• We developed a system that utilize a question-answering system.
Related work

• GeoTime information retrieval may be regarded as a special case of IR4QA.
  – Many approaches to IR4QA introduce some extensions to treat natural sentence questions or question types.
  – Their foundation are information retrieval systems[Sakai et al. 2008].
• There are some text processing method based on the result of question answering system.
  – [Mori et al. 05] proposed a method for multi-answer-focused summarization using a question-answering engine.
    • Importance of each sentence is calculated based on the scores of answer candidates appeared in the sentence.
• Our approach to GeoTime information retrieval takes the same kind of approach as the latter researches.
  – In these researches, the scores of answer candidates are used to weight sentences.
  – In our GeoTime information retrieval, documents are weighted according to the score.
Proposed method

• The proposed method consists of the following three procedures.
  1. Decomposing a complex GeoTime question into a set of simple factoid questions,
  2. Factoid question-answering for the simple questions, and
  3. Scoring documents according to the scores of answer candidates in each document.
System Overview

Decomposition of a complex question

GeoTime question

Factoid question answering Engine

News corpus

Algorithm 3.1
Algorithm 3.3
Algorithm 3.4

AC: Answer candidate
S: Score of AC
D: Document where AC appears

Integration answer candidate in terms of doc and interrogative

Scoring documents

Document with score
Step 1

Question decomposition (1)
Question decomposition (2)

- GeoTime questions are usually complex questions, which have multiple interrogatives, like when, where, etc.

- We suppose that each GeoTime question is able to be decomposed into a set of simple factoid questions.
  - \(<Q, \text{interrog}>\)
    - \(Q\) is a simple question with one interrogative \(\text{interrog}\).

- These simple factoid questions may be handled a factoid question-answering system.

---

Algorithm 3.1: \textsc{DecomposeQuestion}(Qc)

\begin{verbatim}
comment: returns a set of tuples of \((Q, \text{interrog})\), where
\(Q\) is a simple question with one interrogative \(\text{interrog}\), which is obtained by the decompositon of an inputted complex GeoTime question \(Qc\).

global InterrogPats
comment: InterrogPats is a set of patterns that match
with interrogatives in question sentences.

procedure \textsc{PatternMatch}(Str, Pats)
comment: returns a set of tuples of position \((\text{PosS}, \text{PosE})\), where 
\(\text{PosS}\) and \(\text{PosE}\) are the start and end positions of a substring
of \(Str\) matched with one of patterns \(Pats\).

return \(((\text{PosS}, \text{PosE}))\)

procedure \textsc{Substr}(Str, \((\text{PosS}, \text{PosE})\))
comment: returns a sub-string \(\text{SubStr}\) of \(Str\) that starts
from position \(\text{PosS}\) and ends at position \(\text{PosE}\).

return (SubStr)

procedure \textsc{DelSubstrs}(Str, Matches)
comment: returns a string \(Str1\) that is obtained by deleting
all substring expressed by \(Matches\) from a string \(Str\).

return (Str1)

main
\(Ms \leftarrow \textsc{PatternMatch}(Qc, \text{InterrogPats})\)
\(Qs \leftarrow \bigcup_{M \in Ms}\{(\text{\textsc{DelSubstrs}}(Qc, Ms \setminus \{M\}), \text{\textsc{Substr}}(Qc, M))\}\)
return (Qs)
\end{verbatim}
Question decomposition (3)

• Example GeoTime question (GeoTime-0003)
  – いつ、どこでポール・ニッツは、亡くなりましたか？
    • (When and where did Paul Nitze die?)

• Decomposed questions
  1. いつポール・ニッツは、亡くなりましたか？
     • (When did Paul Nitze die?)
  2. どこでポール・ニッツは、亡くなりましたか？
     • (Where did Paul Nitze die?)

• Our current implementation of question decomposition is based on a simple pattern-match.
Decomposition of a complex question

Factoid question answering Engine

When and where …… ?

GeoTime question

Algorithm 3.1

Algorithm 3.3

Algorithm 3.4

Step 2

Obtaining all answer candidates (1)

AC: Answer candidate
S: Score of AC
D: Document where AC appears

News corpus

Integration answer candidate in terms of doc and interrogative

Simple question + its interrogative

AC1, S1 D4 When

AC2, S2 D9 When

AC3, S3 D7 When

AC9, S9 D4 where

AC8, S8 D9 where

AC7, S7 D9 where

D9 S2 + S8

D4 S1 + S9

Document with score

Obtaining all answer candidates (1)
Obtaining all answer candidates (2)

- The algorithm calls a factoid question-answering system to obtain answer candidates and their scores for all of the simple questions.
- \(<D,\text{interrog},AC,Sr,Sw>\)
  - \(D\): document
  - \(\text{Interrog}\): interrogative
  - \(AC\): answer candidate
  - \(Sr\): raw score of \(AC\)
  - \(Sw\): weighted score of \(AC\)

---

**Algorithm 3.3: GETALLANSCANDS(Qs)**

**comment:** returns a set of tuples of \(<D,\text{interrog},AC,Sr,Sw>\), where \(AC\) and \(D\) are an answer candidate and a document in which the answer candidate appears. \(\text{interrog}\) is the interrogative asked in a decomposed question. \(Sr\) and \(Sw\) are the raw and weighted score of the answer candidate. The inputs \(Qs\) is a set of decomposed questions.

**procedure QA(Q)**

**comment:** returns a set of tuples of \(<AC,D,Sr,Sw>\) for the question \(Q\) by using a factoid question-answering system.

\[
\text{return } \{(AC,D,Sr,Sw)\}
\]

**main**

\(ACs \leftarrow \{\}\)

\[\text{for each } (Q,\text{interrog}) \in Qs\]

\[\text{As} \leftarrow \text{QA}(Q)\]

\[\text{for each } (AC,D,Sr,Sw) \in As\]

\[\text{do } ACs \leftarrow ACs \cup \{(D,\text{interrog},AC,Sr,Sw)\}\]

\[\text{return } (ACs)\]
Ordinary search engine based on tf*idf and the vector space model. We do not introduce no feedbacks.
広辞苑 第五版 は いつ 発売 され ました か

KOJIEN 5th-edition TOP when be_on_sale Interrogation

広辞苑 第五版 は いつ 発売 され ました か

Nov.11 at KOJIEN Rel 5th-edition NOM be_published

TC=1, Similarity=2 \(\rightarrow\) Score=3.0

1月11日 に 広辞苑 の 第五版 が 刊行 される

KOJIEN 5th-edition TOP when be_on_sale Interrogation

Nov.11 at KOJIEN Rel 5th-edition NOM be_published

TC=0, similarity=1 \(\rightarrow\) Score=1.0

1月11日 に 広辞苑 の 第五版 が 刊行 される

KOJIEN 5th-edition TOP when be_on_sale Interrogation

Nov.11 at KOJIEN Rel 5th-edition NOM be_published

TC=0, Similarity=2 \(\rightarrow\) Score=2.0

Example of Sentential Matching

1) Each AC is supposed to be an answer

2) Similarity between \(L_Q\) and \(L_i\) is calculated.
Basic factoid-type QA system (3)

Raw score for answer candidates

• **It is difficult to make QA systems high precision with one monolithic method.**
  
  – There is a trade-off relation between informativeness and robustness of analysis in each processing technique.
    
    • More informative ↔ Less informative
    
    • Less robust ↔ More robust
  
• We employ **multiple complementary methods** in order for our QA system to have a variety of informativeness and robustness.

• Implementation: **Raw score** for an answer candidate \( AC \) in the \( i \)-th retrieved sentence \( L_i \) with respect to a question sentence \( L_q \).

\[
S(AC, L_i, L_q) = Sb(AC, L_i, L_q) + Sk(AC, L_i, L_q) + Sd(AC, L_i, L_q) + St(AC, L_i, L_q)
\]

**Bi-gram**  
**keyword**  
**Dependency between an answer candidate and a keyword**
Basic factoid-type QA system (4)

Pseudo voting method in search scheme

• Many existing QA systems exploit global information about answer candidate.
  – **Voting method** --- *boosting the score for answers that occur multiple times* [Clarke 01, Xu 03].

• Pseudo voting [Mori 05]
  – Since our method continues searching for answers until scores of *n different* answers are fixed in n-best search, the system may find *other answer candidates that have same surface expression*.
  – We can *use the partial frequency information* with regard to found answer candidates.
  – **Weighted score** $S^v(AC, L_q)$ for an answer candidate $AC$ is:
    \[
    S^v(AC, L_q) = (\log_{10}(freq(AC, AnsList)) + 1) \cdot \max_{L_i} S(AC, L_i, L_q)
    \]
    where $AnsList$ is the list of answer candidates whose scores are fixed.
Step 3
Scoring documents (1)

First, all answer candidates (ACs) are grouped by document, and then ACs in a document are grouped by interrogative of simple question.
Scoring documents (2)

- All answer candidates (ACs) are grouped by document, and then ACs in a document are grouped by interrogative of simple question.
- We define the **sub-score of document** in terms of an interrogative as the **maximum score of answer candidates that associated with the interrogative**, and finally define the **score of document** as the summation of the sub-scores over all interrogatives.
- Since we have two types of scores of ACs, namely weighted scores and raw scores, two scoring strategies, **Strategy 1 (weighted score)** and **Strategy 2 (raw score)**, are prepared, respectively.

Algorithm 3.4: ScoreDocs(ACs, Strategy)

<table>
<thead>
<tr>
<th>comment</th>
<th>returns a set of tuples of ( (D, S) ), where ( S ) is the score of document ( D ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>procedure Docs(ACs)</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>returns a set of all documents appeared in ACs.</td>
</tr>
<tr>
<td>return</td>
<td>( {D} )</td>
</tr>
</tbody>
</table>

| procedure INTERROGS(ACs) |
| comment | returns a set of all interrogatives appeared in ACs. |
| return | \( \{\text{Interrogative}\} \) |

| procedure ScoreDoc1(D, ACs) |
| return | \( \sum_{i \in \text{INTERROGS(ACs)}} \max_{(D, i, AC, Sr, Sw) \in ACs} Sw \) |

| procedure ScoreDoc2(D, ACs) |
| return | \( \sum_{i \in \text{INTERROGS(ACs)}} \max_{(D, i, AC, Sr, Sw) \in ACs} Sr \) |

main

\( DSs \leftarrow \{} \)

for each \( D \in \text{Docs(ACs)} \)

if \( \text{Strategy} == 1 \)

then \( DSs \leftarrow DSs \cup \{(D, \text{ScoreDoc1}(D, ACs))\} \)

else if \( \text{Strategy} == 2 \)

then \( DSs \leftarrow DSs \cup \{(D, \text{ScoreDoc2}(D, ACs))\} \)
Experimental Results (1)

Settings

• We conducted four runs shown in Table 3.
• The difference among the runs is due to:
  – Scoring strategy and
  – Parameter settings of the question-answering system.

<table>
<thead>
<tr>
<th>Table 1: Description of system parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>a: Number of answers to be searched.</td>
</tr>
<tr>
<td>d: Number of documents to be retrieved.</td>
</tr>
<tr>
<td>ppd: Maximum number of passages retrieved from one document.</td>
</tr>
<tr>
<td>p: Number of passages to be considered in the retrieved documents.</td>
</tr>
<tr>
<td>pwin: Number of sentences in one passage.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Common parameter settings of the question-answering system</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>250</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3: Submitted runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run ID</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>FORST-JA-JA-01-D</td>
</tr>
<tr>
<td>FORST-JA-JA-02-D</td>
</tr>
<tr>
<td>FORST-JA-JA-03-D</td>
</tr>
<tr>
<td>FORST-JA-JA-04-D</td>
</tr>
</tbody>
</table>
Experimental Results (2)

Overall evaluation

• Strategy 2 (raw score) is superior to Strategy 1 (weighted score).
  – In GeoTime retrieval, documents with answer candidates for both ‘when’ and ‘where’ are important.
  – The weighted score scheme may give wrongly high value to documents that have only one kind of answer candidates.
  – We need more detailed analysis.

• The parameter settings of question answering do not seriously affect to the effectiveness in GeoTime retrieval.

• There are no statistically significant difference among runs according to the Wilcoxon matched pairs signed rank sum test.

<table>
<thead>
<tr>
<th>Run ID</th>
<th>mean AP</th>
<th>mean Q</th>
<th>mean nDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORST-JA-JA-01-D</td>
<td>0.233</td>
<td>0.259</td>
<td>0.332</td>
</tr>
<tr>
<td>FORST-JA-JA-02-D</td>
<td>0.286</td>
<td>0.284</td>
<td>0.372</td>
</tr>
<tr>
<td>FORST-JA-JA-03-D</td>
<td>0.206</td>
<td>0.238</td>
<td>0.324</td>
</tr>
<tr>
<td>FORST-JA-JA-04-D</td>
<td>0.276</td>
<td>0.287</td>
<td>0.377</td>
</tr>
</tbody>
</table>

Table 3: Submitted runs

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Strategy</th>
<th>a</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORST-JA-JA-01-D</td>
<td>1 (weighted score)</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>FORST-JA-JA-02-D</td>
<td>2 (raw score)</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>FORST-JA-JA-03-D</td>
<td>1 (weighted score)</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>FORST-JA-JA-04-D</td>
<td>2 (raw score)</td>
<td>20</td>
<td>60</td>
</tr>
</tbody>
</table>

a: Number of answers to be searched.
p: Number of passages
Experimental Results(3)

Per-topic evaluation

- There are some topics that cannot be appropriately handled by our method.
  - The method lacks in robustness in terms of variety of queries.
- Especially, the question decomposition module failed to decompose GeoTime questions in some cases as follows.
Failure analysis (1)
in question decomposition

• Failures because of lack of patterns.
  – GeoTime-0010: いつITERの設置とその建設予定地が決定しましたか？
    • (When was the decision made on siting the ITER and where is it to be built?)
  – GeoTime-0018: 2002年に合衆国がある国に侵攻したのは何月何日でしたか？
    • (What date was a country was invaded by the United States in 2002?)
Failure analysis (2)
in question decomposition

• Failures because the given questions consist of two separate questions.
  – They cannot be handled by our question-answering systems.
  – We need a system for information access dialogue (IAD) task like NTCIR-5 QAC.
  – GeoTime-0015:どのアメリカンフットボールチームが、2002年のスーパーボウルで優勝しましたか、また、試合はどこで開催されましたか？
    • (What American football team won the Superbowl in 2002, and where was the game played?)
  – GeoTime-0020:もっとも最近に国連に加盟したのはどの国ですか、また、加盟したのはいつですか？
    • (What country is the most recent to join the UN and when did it join?)
  – GeoTime-0023:欧州連合の最大の規模拡大が生じたのはいつですか、また、どの国がメンバーになりましたか？
    • (When did the largest expansion of the European Union take place, and which countries became members?)
Conclusion

• We proposed a method of GeoTime information retrieval based on question decomposition and question answering.
• We demonstrated that the proposed method is able to accept GeoTime questions and retrieve relevant documents to some extent.
• However, there is still room to improve the effectiveness of retrieval.
  – Question decomposition, etc.
Thank you very much!!