A Monolithic Approach and a Type-by-Type Approach for Non-Factoid Question-answering

Tatsunori MORI, Mitsuru SATO, Madoka ISHIOROSHI,
Yugo NISHIKAWA, Shigenori NAKANO and Kei KIMURA

Graduate School of Environment and Information Sciences

Yokohama National University

mori@forest.eis.ynu.ac.jp

Introduction and Related work

- Probabilistic model for definitional question-answering [Han et al. 06]
  1. General language model
  2. Topic language model: relevance of answer candidate to the question.
  3. Definition language model: appropriateness of writing style in terms of definition.
- Non-factoid QA can be modeled from at least two aspects.
- Most of previous studies follow the same scheme
  - Lexico-syntactic patterns to find answer candidates in terms of the appropriateness of writing style [Fujii 02, Morooka 06]
  - Method to extract from Wikipedia a list of interest-marking terms and use them to score answer candidates [Kosseim et al. 05]
  - Summarization techniques are used to reduce redundancy in answer candidates [Blair-Goldensohn 03]

Our approach

- As described by Han et al. [Han 06], we also hypothesize that the appropriateness of answer candidate for non-factoid questions can be measured by the combination of two measures:
  - *Appropriateness of writing style*: how appropriate is the writing style of the candidate in terms of the type of a give question
  - *Relevance to the topic of question*: how relevant is the candidate to the "topic" of the question.

Our contribution

- We basically adopts major techniques in the previous studies:
  1. Lexico-syntactic patterns to find key sentences,
  2. Passage retrieval, and
  3. Summarization based on passage clustering.
- Main contribution of this study
  1. Query expansion using the Web to improve the coverage of retrieval of documents related to the topic of question.
  2. Variable-length passage retrieval based on lexical chains.
  3. Method to find interest-marking words and eliminate unrelated chains, in order to detect only topic-related lexical chains.
  4. Unified score for answer candidates calculated from document score, passage score, and score of writing style.
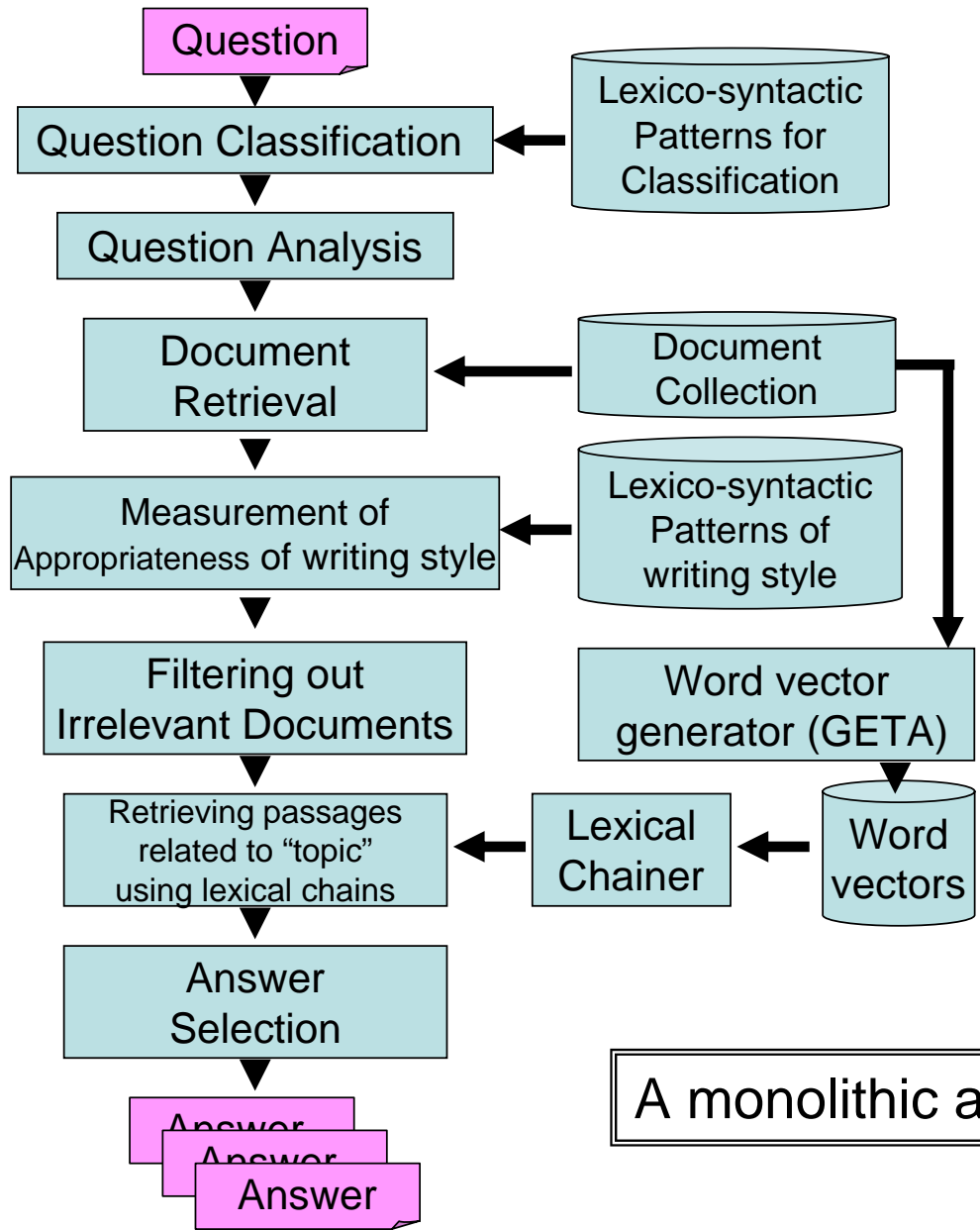
4

Two systems for NTCIR-6 QAC (QAC4)

- A system with a *monolithic* architecture (ID = forst1)
  - Processes all types of questions including factoid questions uniformly.
    - A set of manually constructed lexico-syntactic patterns.
    - A method to retrieve passages related to topic of question by using lexical chains.
- A system with a *type-by-type* architecture (ID = forst2)
  - Four specialized subsystems
    - For definitional and other type question
    - For why-type question
    - For how-type question
    - For factoid question
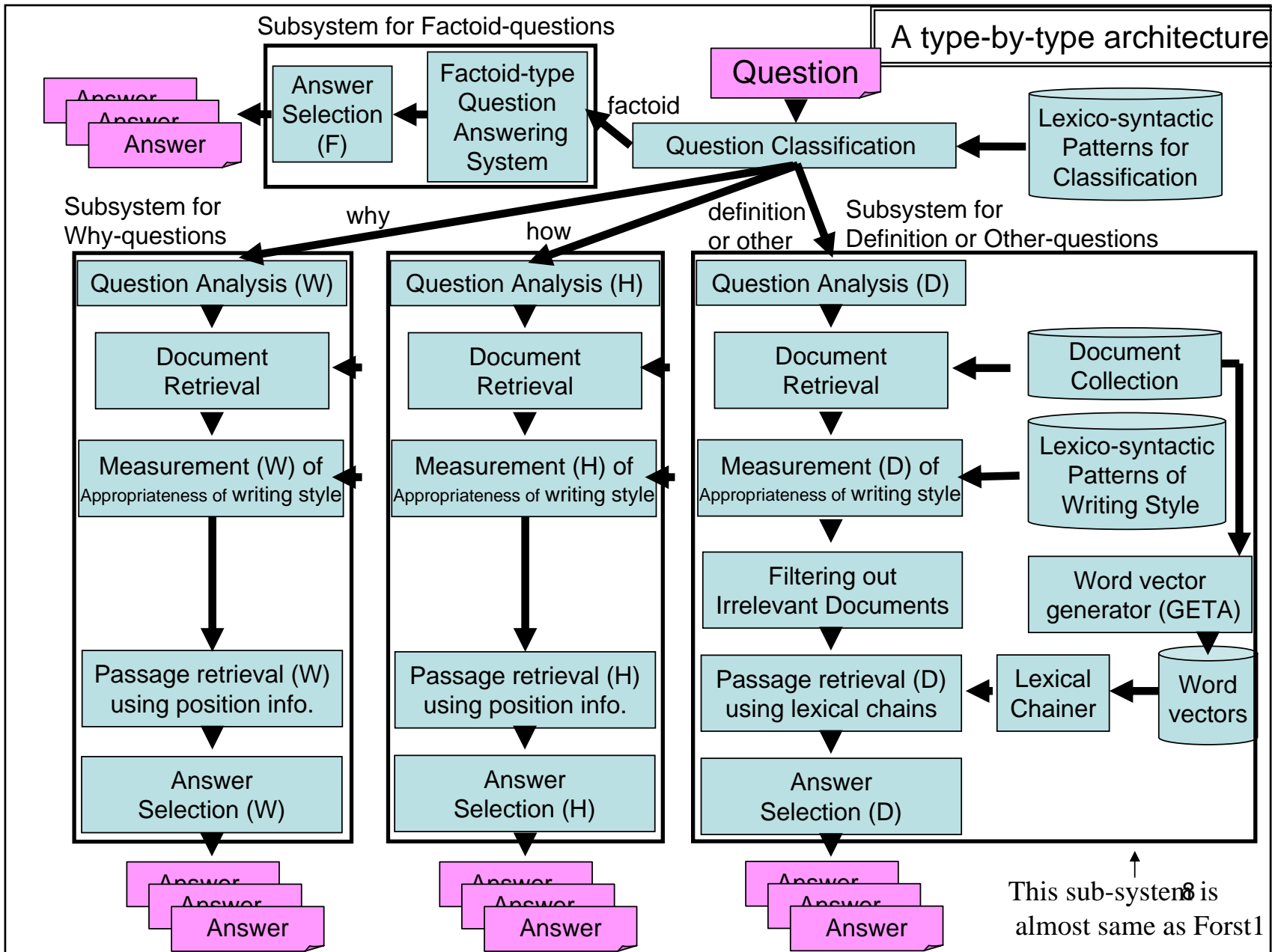 - Given questions are dispatched to one of the subsystems by module of question classification.

Types of question

Q. type	Examples
definition	"...-to-wa nan-desu-ka", "...-tte nani" (what is ...)
why	" naze ...-desu-ka" (why ...) "... riyuu-wa nan-desu-ka" (what is the reason for ...)
how	"... dou su-reba yoi-desu-ka" (how do I do ...) "... yari-kata-wa nan-desu-ka" (what is the way to do ...)
factoid	"...-wa dare-desu-ka" (who is ...) "...-wa doko-desu-ka" (where is ...)
other	"...-wa dou-desu-ka" (how about ...)

➢ Questions classified into *factoid* are treated as *other* in the system with a monolithic architecture (i.e. forst1).

A monolithic architecture

A type-by-type architecture

Subsystem for Factoid-questions

Question → Question Classification

Question Classification → factoid → Factoid-type Question Answering System → Answer Selection (F) → Answer / Answer / Answer

Question Classification ← Lexico-syntactic Patterns for Classification

Subsystem for Why-questions

why

Question Analysis (W) → Document Retrieval → Measurement (W) of Appropriateness of writing style → Passage retrieval (W) using position info. → Answer Selection (W) → Answer / Answer / Answer

how

Question Analysis (H) → Document Retrieval → Measurement (H) of Appropriateness of writing style → Passage retrieval (H) using position info. → Answer Selection (H) → Answer / Answer / Answer

Subsystem for Definition or Other-questions

definition or other

Question Analysis (D) → Document Retrieval → Measurement (D) of Appropriateness of writing style → Filtering out Irrelevant Documents → Passage retrieval (D) using lexical chains → Answer Selection (D) → Answer / Answer / Answer

Document Retrieval ← Document Collection

Measurement (D) of Appropriateness of writing style ← Lexico-syntactic Patterns of Writing Style

Document Collection → Word vector generator (GETA) → Word vectors → Lexical Chainer → Passage retrieval (D) using lexical chains

This sub-system is almost same as Forst1

8

Detailed Techniques

- Document retrieval with query expansion using Web (described later)
- The type-by-type approach (forst2)
  - Definition-type and Other-type questions are processed by the almost same system as forst1.
  - Subsystem for why-type questions
    - Retrieves fixed length passages (3 sentences)
    - Utilize *relative position of the passage* to the key sentence
  - Subsystem for how-type questions
    - Detection of *focused predicate*.
    - *Paraphrasing relative clause* with the focused predicate in order *to dynamically generate lexico-syntactic patterns* of writing style.
- The monolithic approach (forst1, described later)
  - Score of key sentences in terms of writing style
  - Passage retrieval using lexical chains
  - Answer selection: scoring and redundancy reduction

9

Document retrieval
with query expansion using Web
to improve the coverage of topic-related documents

1. Retrieve **top L snippets from a Web search engine** by submitting keywords.

2. Give a "word score" to each word in snippets, and select top M words as **additional keywords**.

$$score_{word}(w) = freq_{snippet}(w) \cdot IDF_{DB}(w)$$

3. Perform document retrieval with a in-house tf*idf/VSM-based search engine in three settings

   a. Ordinary search with the original keywords

   b. "AND" search with the original keywords

   c. Ordinary search with original keywords with the weight 1.0 and additional keywords with the weight 0.5

4. Merge retrieved documents and give a normalized **document score** to each document:

$$score_{doc}(D_i) = \frac{w(D_i) \cdot sim(Q, D_i)}{\max_d \{w(d) \cdot sim(Q, d)\}} \qquad w(D_i) = \begin{cases} 1.0 & \text{obtained by (a)} \\ 2.0 & \text{by (b) or (c)} \end{cases}$$

Answer candidate and its score

- Answer candidate = passage
- Scoring answer candidates

$$
\begin{aligned}
score_{AC}(AC) &= \alpha \cdot score_{doc}(doc(AC)) \\
&+ \beta \cdot score_p(AC) \\
&+ \gamma \cdot score_{AC}^{style}(AC)
\end{aligned}
$$

Document score

Passage score

Score of writing style

Retrieved document

Passage relevant to the topic of question

Appropriate writing style

Key Sentence

Answer Candidate (AC) = *passage*

11

Score of appropriateness in terms of writing style

- Lexico-syntactic patterns that represent the clue of anwers for each question type.
- Each pattern has a score of appropriateness ([0,1]), which is assigned to matched sentences.
- Definition-type and how-type: each pattern is generated from one of **templates** that has a **slot** which will be filled by a **question target**.
  - Examples

    Definition-type: **<target> ? .+<QFocus>**

              (**<target> is a <QFocus> that .+**)

    Why-type: **<focusedPred> ? .+ <ReasonExp>**

              (**.+ <focusedPred> .+, <ReasonExp> .+**)

- How-type and other-type: instead of patterns, it is examined whether **compound nouns** or **focused predicates** in the question appear in the sentences.

12

Passage retrieval using lexical chains (1/2)

- Lexical chain
  - A sequence of semantically related words in a text.
  - Used for capturing a portion of the cohesive structure of the text.
  - We utilize it to retrieve passages related to the topic of question.

- Lexical chainers [Mochizuki 00]
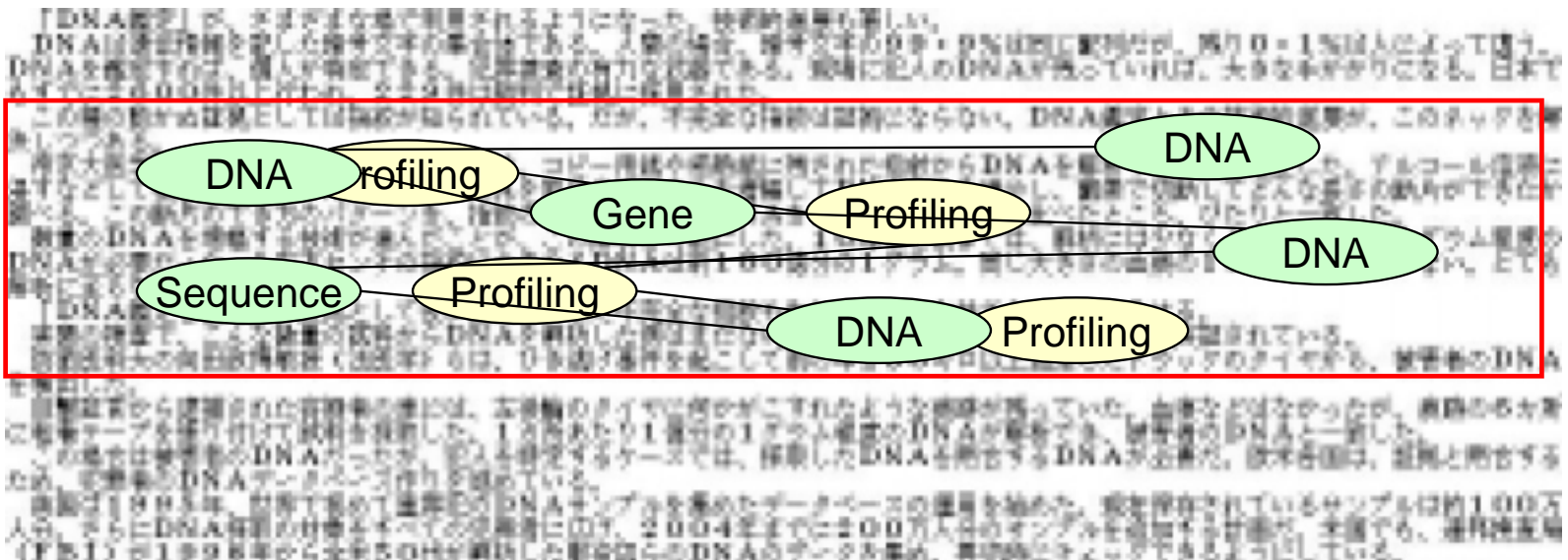
Passage retrieval using lexical chains (2/2)

- Mochizuki's method [Mochizuki 00] detects a portion of text as a passage if all sentences of the portion are covered by **at least one lexical chain**.
  - The method tends to extract **longer passages**.

- Our method: more moderate
  1. Calculate "density of lexical chain" for each sentence, and select sentences whose density are maximal. (The centers of passages S_c)
  2. Determine the extent $(S_s, ..., S_e)$ of passage for each center S_c of passage according to the density of sentences adjacent to the center.

$$s = \min\{j | j \leq \forall k \leq c$$
$$(density_{win}(S_k, S_c) \geq R_d \cdot density(S_c))\}$$

$$e = \max\{m | c \leq \forall k \leq m$$
$$(density_{win}(S_k, S_c) \geq R_d \cdot density(S_c))\}$$

$$density_{win}(S_j, S_c) = density(S_j) \cdot \frac{L^2 - |j-c|^2/2}{L^2}$$

14

Answer selection (1/2)

- Scoring answer candidates

$$\begin{aligned}
score_{AC}(AC) &= \alpha \cdot score_{doc}(doc(AC)) \\
&+ \beta \cdot score_p(AC) \\
&+ \gamma \cdot score_{AC}^{style}(AC)
\end{aligned}$$

Document score

Passage score

Score of writing style

AC
(*Answer Candidate*)

Retrieved document
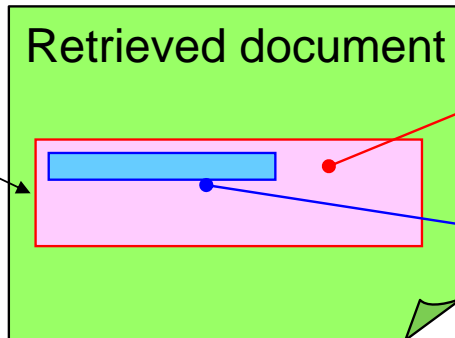
Passage relevant to the topic of question

Appropriate writing style

- Redundancy control by clustering-based summarization

Answer Selection (2/2)
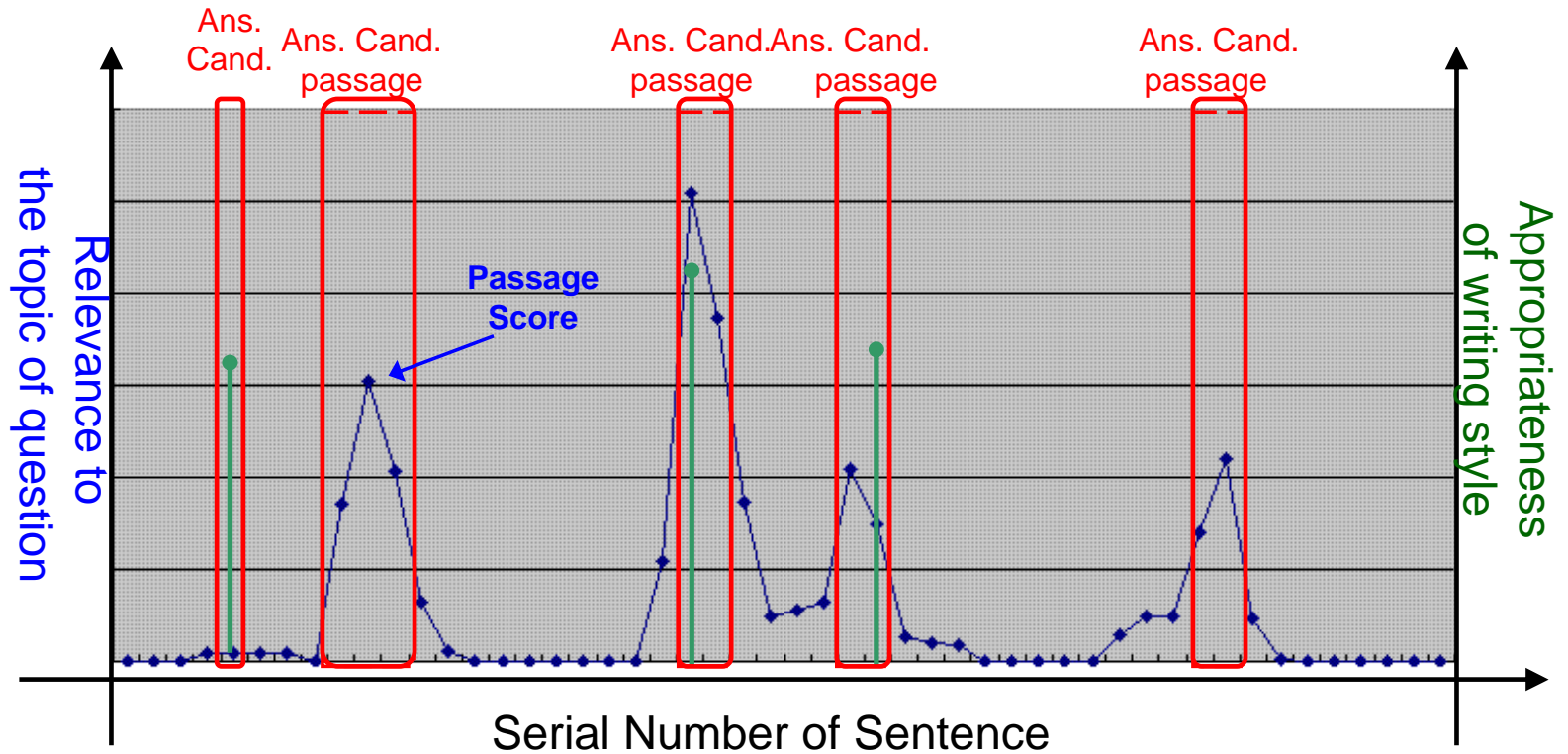
Runs at NTCIR-6 QAC

- forst1: a monolithic architecture
- forst2: a type-by-type architecture

Number of answers with judgment

✓ Micro average of precision.
✓ Performance overview in terms of precision

Legend:
- A
- B
- C
- D
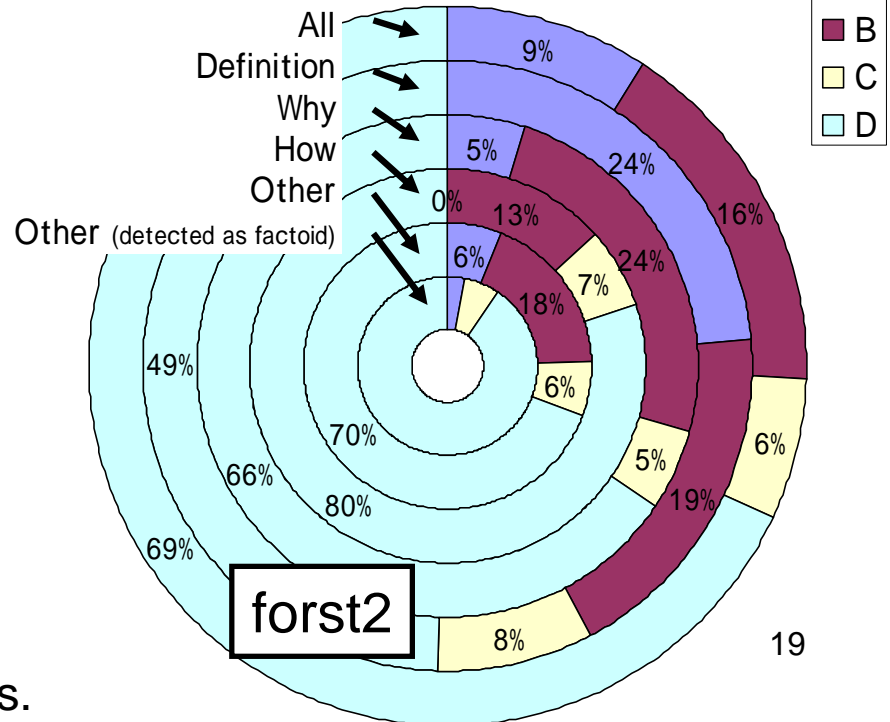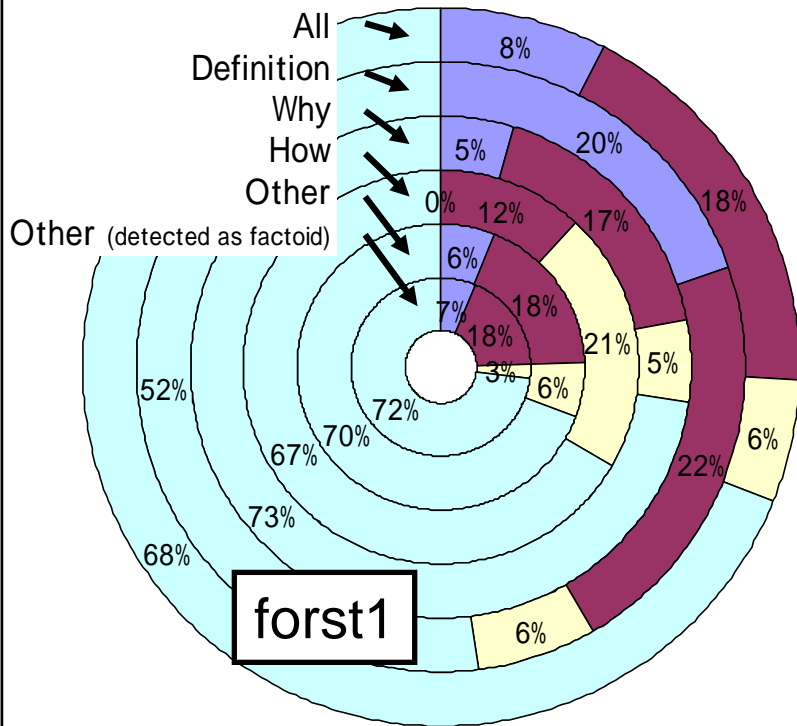
- Both systems have better performance for definitional Qs.
- There are more answers in B judgment than C. The systems return longer answers than expected.
- While forst1 is better for how-type Qs, forst2 is better for why-type Qs.

19
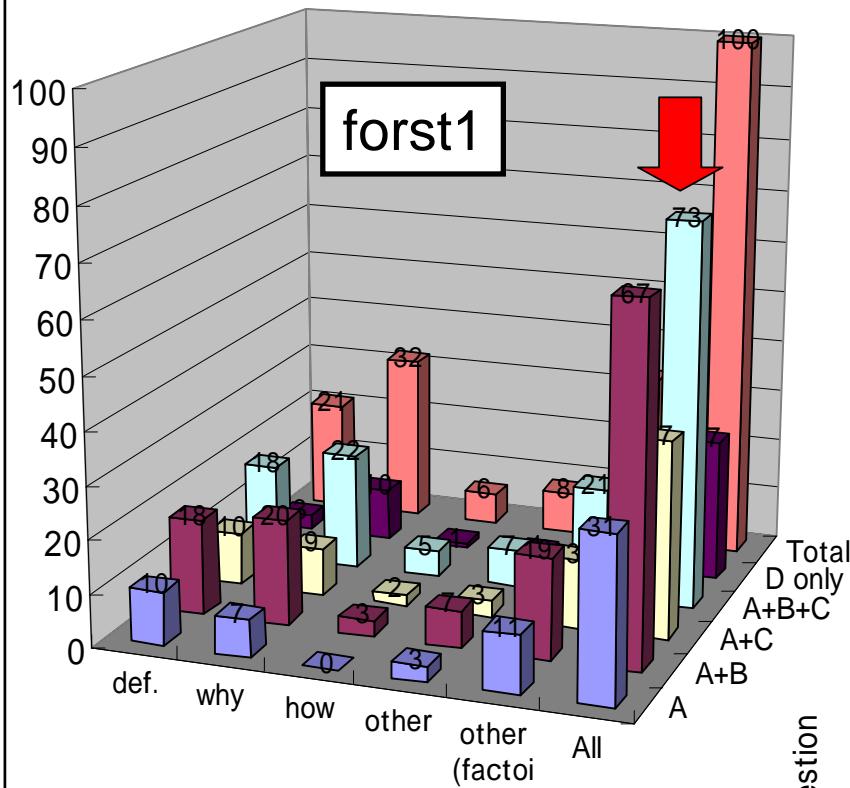
Number of questions with judgment

forst1

✓ Performance overview in terms of question coverage

forst2

- Forst1 and forst2 found at least one partially correct answer for 73% and 50% of Qs, respectively.
- The main reason of difference of two systems lies in the questions detected as factoid. It is caused by the failure of Q. classification.
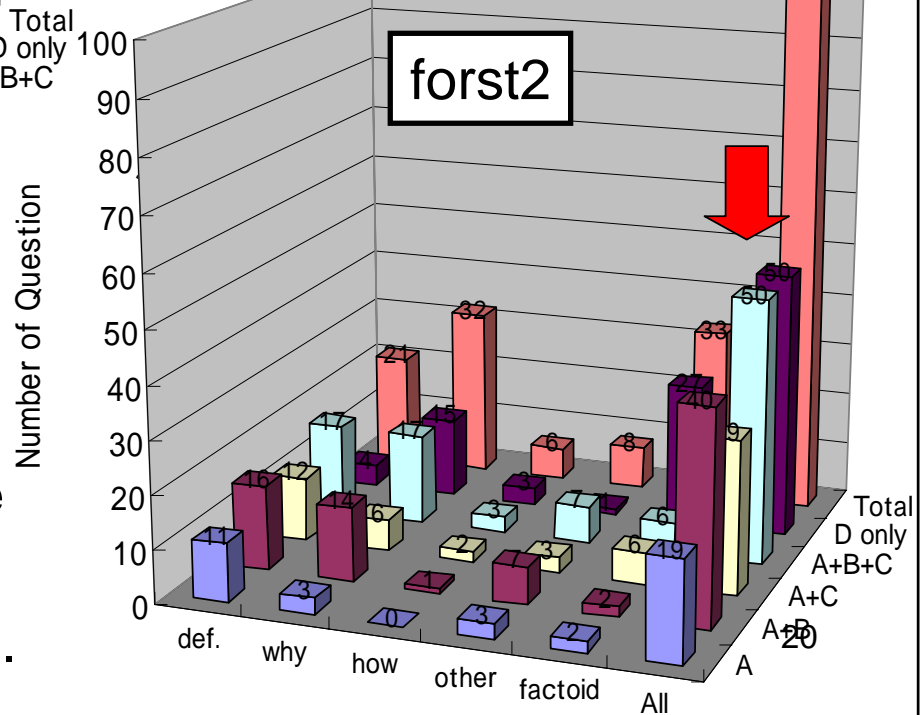
MRR
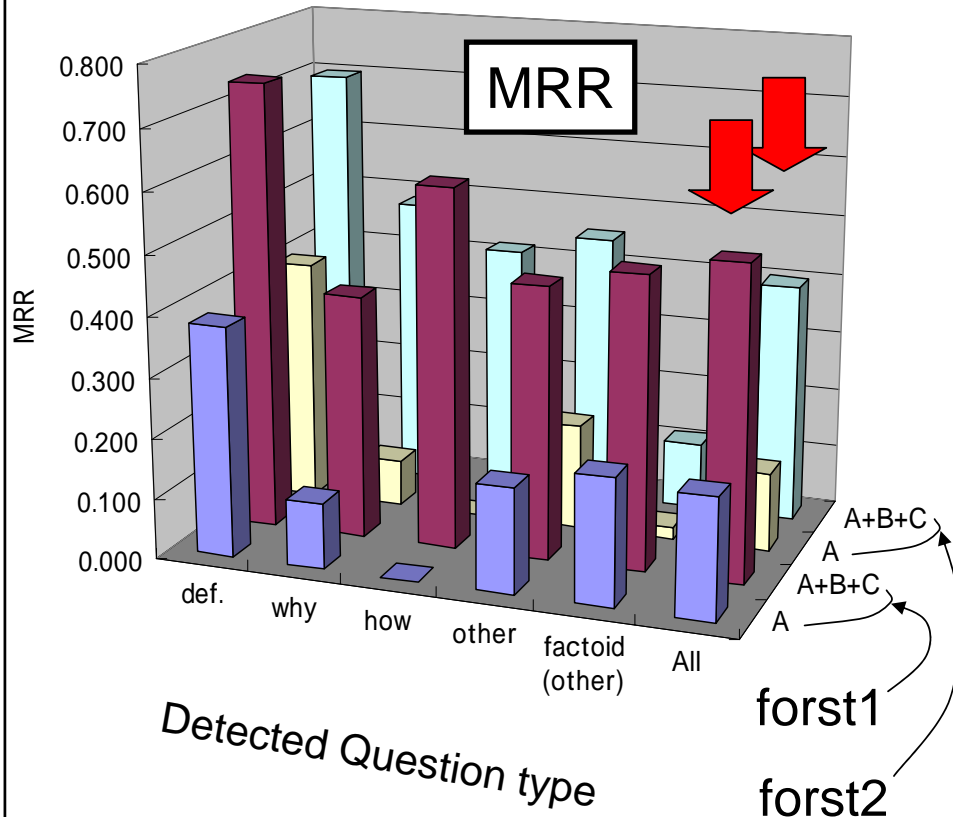
- ✓ Ability of ranking answer candidates (i.e. scoring methods)

- MRR values are 0.517 (forst1) and 0.400 (forst2).

- Forst1 produces generally better ranking than forst2

21

Performance of question classification

- Many of non-factoid questions are incorrectly classified into the type of factoid.

- While forst2 is affected by the failure, forst1 is not affected because it treats all questions as non-factoid type.

- In actual situation, we have to have better question classification.

(What is the wiretap law?)

- Forst1
  - A

  - B   ...   ...

  - C

  - D

(What is the wiretap law?)

- Forst2
  - A

  - B

  - C

  - D

QAC4-00010-00:
(Why is an airport necessary in Kobe?)

- Forst1
  - A

  - B

  - C

  - D

QAC4-00010-00:
(Why is an airport necessary in Kobe?)

- Forst2
  - A

  - B

  - C

  - D

Concluding remarks

- Two types of approaches
  - A monolithic architecture
    - retrieves answer passages related to a question using lexical chain
  - A type-by-type architecture with four subsystems
- The former approach has better performance than the latter.
  - ✓ Passage retrieval using lexical chains and the scoring function work well.
  - ✓ But the failure of the latter approach is mainly due to the failure on the question-type classifier.