

## Open issues and future directions of interactive QA

Speaker: Boris van Schooten

University of Twente,  
Human Media Interaction group

Netherlands

## Human Media Interaction group

- about 50 members
- topics:
  - dialogue systems
  - language technology
  - information retrieval
  - AI, multimedia systems, virtual reality, HCI
- Interactive QA projects:
  - IMIX (Vidiam + Imogen), finished since end 2007
  - MESH, just started

## Contents

### Summary:

- Current state of interactive QA
- Handling follow-up questions
  - a comparison of IR and NLP strategies
  - towards a broader theory of follow-up utterances
  - information need
- What can be done with system utterances
- Working with real users
  - obtaining corpora
  - user learning

## Question Answering: introduction

- QA aims to give precise answers to natural language questions, by searching an unstructured set of documents.
- QA is similar in spirit to search engines.
  - helps in searching large document databases
  - knowledge poor approach
- Main differences:
  - QA tries to use the information in full sentences
  - QA returns answers, not documents

## Question Answering: introduction

- example: factoid, open domain

*Q: What is the capital of France?*

*A: Paris*

- example: non-factoid, specific domain

*Q: What do I have to do to prevent RSI?*

*A: In the case of prolonged computer work, you are advised to take regular breaks and do exercises.*

## Question Answering: introduction

- Main tasks in QA process:

- question analysis

extract keywords, answer type, etc. from a natural language parse of the question

- document retrieval

similar to search engines; rank documents by relevance to analysed question

- text fragment selection and processing

select the relevant text fragments from the most relevant documents, possibly count them or combine them to form the final answer (or answers)

## Question Answering: introduction

### Interactive QA

- add interactivity to QA, in the form of natural language dialogue with the purpose of enabling iterative search refinement, by:
  - enabling users to pose questions in context of previous dialogue (follow-up questions, FQ)
  - enabling users to give feedback on the given answers (follow-up utterances, FU)
  - asking counter questions or giving suggestions to the user

## Question Answering: introduction

- Traditional TREC-like FQ:

*Q1: Who is the filmmaker of the Titanic?*

*Q2: Who wrote the scenario?*

*Q3: How long did the filming take?*

- Real-life FU (IMIX corpus):

*Q2: So the answer is 'no'?*

*Q2: What is meant here by 'hygienic handling'?*

*Q2: Is this always true?*

*Q2: One minute per what?*

## State of the art in Interactive QA

- Limited research field; only about 10 QA systems with some dialogue functionality
- Much is based on a setup like the Trec/Qac context task. Assumptions:
  - text-based QA
  - follow-up utterances (FU) are factoid questions with a unique answer
  - FU are based on correct previous answer
  - no topic shifts

## State of the art in Interactive QA

- So, this setting does not account for:
  - non-factoid questions
  - asking questions about the system's answer
  - actual user utterances
  - different user information needs
  - steering the IR process by means of dialogue
  - speech or multimodality
- We tried to cover some of this ill-charted area in VIDIAM. Work has been done in both the IMIX and RITEL projects.

## IR strategies for FQ

- All IR strategies in the literature are one of:
  1. question rewriting  
IMIX, Rits-QA, KAIST
  2. adding keywords from previous utterances  
Ritel, ASKA, NTU system
  3. searching in previous search results  
De Boni, OGI system
- Third strategy actually seems to have the best performance!

## IR strategies for FQ: question rewriting (IMIX)

### 1. Build referent database

Q1: What is RSI? -> { RSI }

A1: RSI is also known as mouse arm. -> {RSI, mouse arm}

Q2: How can I prevent it?

### 2. Identify type of rewriting required

replace anaphor / add PP / expand ellipsis

### 3. Identify anaphors (and other features)

Q2: How can I prevent it?

### 4. Rewrite

Q2: How can I prevent RSI?

## IR strategies for FQ: adding keywords from previous utterances (Ritel)

U1: *On what continent lies Botswana ?*  
(ASR: *on what continent lies {fw} the*)  
**conf.context:** kw = {continent, lies} qtype = location  
**FQ context:** (empty)

S1: You are looking for a location, but what precisely do you want to know?

U2: *The continent of Botswana.*  
(ASR: *the continent of Botswana*)  
**conf.context:** kw = {continent<sub>confirmed</sub>, lies, Botswana}  
qtype=location<sub>confirmed</sub>  
**FQ context:** (empty)

S2: You are looking for a location related to continent and Botswana. I think it's South Africa. Your next question?

U3: *What is its land area*  
(ASR: *what is its land area*)  
**conf.context:** kw = {land area} qtype = number  
**FQ context:** kw = {continent, lies, Botswana, S.Africa}

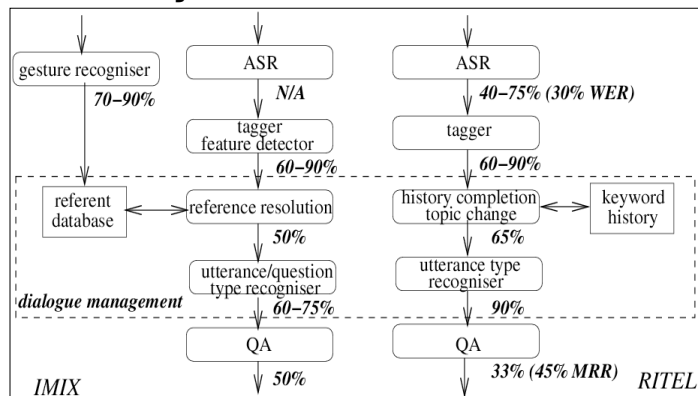
## IR strategies for FQ: typical stepwise approach

What do we need? For the three IR strategies:

- Is the question an FQ?  
required for all strategies
- What referents does the FQ refer to?  
required for adding search terms and rewriting
- How should the FQ be rewritten?  
required for rewriting

And: search in last documents or do new search?  
depends on match between document structure and information need

## IR strategies for FQ: system architectures



## IR strategies for FQ: NLP for question parsing

- Simple topic detection by general keywords
  - Ritel: topic keyword triggers topic shift handling
- Compare keywords with expected keywords for topic argument completion
  - Ritel: hand-coded rules
  - IMIX: failed to obtain reliable results using available NLP modules (too noisy)
- Classify syntactic or thematic relationship between FQ and previous dialogue
  - IMIX: classify referring sentence by syntactic class

## IR strategies for FQ: NLP for QA/IR

- Phrase tagging  
(Ritel, Imix/Rolaquad)
- Intra-sentential relationship tagging  
(Imix/Rolaquad, Imix/QADR)
- Inter-sentential relationship tagging  
(Rare in QA systems. Is used in automatic summarisation)
- Document tagging  
(Hitqa)

## IR strategies for FQ: practical performance

- Some conclusions:
  - Major performance bottlenecks: ASR, reference resolution, rewriting, QA
  - Rewriting is not possible most of the time
  - topic shift detection conceptually flawed, but users do like to mention topic
  - “Search in last document” scores fairly well, and very well for certain types of questions
  - Improve speech recognition by encouraging anaphors and repetition

## A broader theory of follow-up utterances

- Different kinds of user utterances:
  - questions
  - feedback to a previous answer
  - attempts to comprehend or verify a previous answer
  - need to expand the answer to include obviously missing information (discourse question)
  - need for clarification of technical details (turned out to be difficult to distinguish)
  - user-driven versus answer-driven FQ?

## A broader theory of follow-up utterances

- new question (topic shift; do not include context)
- self-contained FQ (free to add context)
- regular FQ (requires context)
  - subclass by context completion strategy
- discourse FQ (show more of context)
  - show missing text
  - show general info about picture (multimodal)
  - show info about visual element (multimodal)
- verify or comprehend answer
- positive/negative feedback:
  - good answer
  - bad answer
  - user signals system did not understand user (speech)

## A broader theory of follow-up utterances

- Are specific IR strategies better for specific types of FQ?
  - We found that “search in last document” is particularly effective for discourse questions (75% versus 35% average)
  - Are there other relationships between FQ type and optimal IR strategy?
    - for example, is there a difference between user-driven and answer-driven FQ?

## A broader theory of follow-up utterances

- Discourse questions in relation to answer category

previous answer	factoid only	factoid w/ context	encyclo- pedic	encyclopedic, incorrect	multimodal
ans. occurrence	9%	16%	75%		
discourse FQs	12%	13%	9%	20%	46%

- Other studies: paragraph-sized answers are preferred
- Relationship between summarisation and interactive QA: anticipating information need.

## Document structure and information need

- Documents are meant for a particular audience, and the author probably has a particular set of questions in mind.
- Example: encyclopedia, each document describes the definition of a particular concept.
- Hypothesis: a document can be understood as a sequence of answers to a logically coherent series of questions.

## Document structure and information need

There is a gardening revolution going on.

People are planting flower baskets with living plants,

mixing many types in one container for a full summer of floral beauty.

To create your own "Victorian" bouquet of flowers,

choose varying shapes, sizes and forms, besides a variety of complementary colors.

Plants that grow tall should be surrounded by smaller ones and filled with others that tumble over the side of a hanging basket.

## Document structure and information need

There is a gardening revolution going on.

*What kind of revolution?*

People are planting flower baskets with living plants,

*What kind of plants?*

mixing many types in one container for a full summer of floral beauty.

*To create your own "Victorian" bouquet of flowers, (= How do I create ...?)*

choose varying shapes, sizes and forms, besides a variety of complementary colors.

*How do I arrange the flowers?*

Plants that grow tall should be surrounded by smaller ones and filled with others that tumble over the side of a hanging basket.

## Document structure and information need

Inter-sentence analysis not done in QA, but is done in summarisation:

- Coreference resolution
  - use traditional syntax/rule-based algorithm
- Statistical semantic relationships using keyword frequencies
- RST (rare)
  - elaboration, motivation, purpose, evidence, condition, contrast

## Document structure and information need

- In IMIX, limited experiments were done with RST-like structures for both summarising and answering FQ. Problems:
  - Annotation labour-intensive, we would need automatic annotation
  - Low inter-annotator agreement of RST annotations
  - Difficulties measuring inter-annotator agreement
  - Multiple viable alternatives

## Task and information need

Q1: *What is the capital of France?*

Q2: *Who is its mayor?*

versus:

Q1: *What is the length of the Mississippi?*

Q2: *And that of an average car?*

The latter type can be a result of execution of a special task. Can task and document structures be matched (find a special document type for a special task type)?

## Task and information need

- We may expect that task shapes information need.
- We did not account for this (we assumed more or less the same task; our users did not really have a task)
- In QA contests, a distinction is made between browsing and gathering FQ. "Gathering FQ" is staying on topic.
- What is the relation between topic shifts, thematic relationships, task, and IR strategy?

## Task and information need

- We are looking at suitability of medical protocols as documents (semi-structured QA):
  - handling FQ and user feedback in a task-based domain (including step-by-step protocols, and under what conditions to do what)
  - inter-sentential semantic structuring of semi-structured documents, including task-oriented structuring

## System utterances: steering the user

- Steer users towards posing easier questions
  - give instructions when questions are wrong.
    - "too long"
    - "not specific enough"
    - "please reformulate"
    - others?
  - Give suggestions for alternative or follow-up questions
  - Give the proper instructions so that questions are of the right type, format, structure, etc.

## System utterances: steering the user

- Can users be taught to ask only factoid (or other limited) questions?
  - has been tried, with surprisingly varying results:
    - Kato et al: tried to steer users by explaining what a factoid question is: it didn't work: 34% non-factoid questions
    - Ritel: tried to steer users by giving examples (and actually telling them they may ask anything they like). It did work: only 12% non-factoid questions
  - So, it appears that giving examples works, but more research is needed.
  - How to instruct users in limited time?

## System utterances: steering the user

- Can users be taught to pose more natural questions?

Affordance: influencing the perceived space of possibilities through appearance.

- Preliminary results indicate a positive effect of the presence of a talking face, and using speech rather than text, on length of utterance
- Talking face: Pilot study was done using IMIX, working on a new study within the MESH project
- Speech: Ritel versus Imix, worth further study

## Corpora: towards a set of guidelines

- How to design (WOz) experiments:
  - How to ensure that the wizard can promptly answer all questions.
    - wizard is domain expert / wizard is trained / wizard uses tailored IR environment / wizard replaces specific submodules?
  - How to make sure that answering performance / error rates have a desirable level.
    - determine desired error rate; introduce artificial errors. Use different error rates and compare result?
  - What about non-factoid answers? Account for difference between human and system answer.

## Corpora: towards a set of guidelines

- How to design scenarios:
  - How to ensure desirable mapping of information need to questions
    - “browsing” scenario / search task (find n items related to ...) / write a summary / answer a specific set of questions / use a real-life setting
  - How to ensure correspondence to real-life setting (real users with real information need).
    - use a real-life setting / analyse human-human dialogues / interview domain experts / use general usability technique

## User learning

- Needs of more experienced users.

*What happens when naïve users will necessarily become expert users?*

  - more emphasis on efficiency, less on learnability
  - more potential for interactive QA
  - will the natural language paradigm compete with more traditional approaches to interaction? Like:
    - “narrow/broaden search” button
    - “new question” button
    - “show more answers” button

## User learning

- Use user feedback to steer the IR process.

*Will users understand concepts of the underlying IR engine?*

- change IR parameters directly in Ritel:
  - add or delete keywords directly
  - directly determine question type
- select between candidate answers (and otherwise manipulate answer/document set)
- what about steering the underlying NLP and search strategies?

## References

- *Follow-up question handling in the IMIX and Ritel system: a comparative study*, JNLE 2007 (forthcoming)
- *Handling speech input in the Ritel QA dialogue system*, Interspeech 2007
- *Follow-up utterances in QA dialogue*, TAL 46(3), 2007