Natural Language Engineering **1** (1): 1–23. Printed in the United Kingdom

Follow-up question handling in the IMIX and Ritel systems: a comparative study

B. W. VAN SCHOOTEN, and R. OP DEN AKKER,
Human Media Interaction,
University of Twente, Netherlands.

S. ROSSET, O. GALIBERT, A. MAX, and G. ILLOUZ,
Spoken Language Processing Group (TLP),
CNRS-LIMSI, France.

(*Received 20 September 2007*)

Abstract

One of the basic topics of QA dialogue systems is how follow-up questions should be interpreted by a QA system. In this paper, we shall discuss our experience with the IMIX and Ritel systems, for both of which a follow-up question handling scheme has been developed, and corpora have been collected. These two systems are each other's opposites in many respects: IMIX is multimodal, non-factoid, black-box QA, while Ritel is speech, factoid, keyword-based QA. Nevertheless, we will show that they are quite comparable, and that it is fruitful to examine the similarities and differences. We shall look at how the systems are composed, and how real, non-expert, users interact with the systems. We shall also provide comparisons with systems from the literature where possible, and indicate where open issues lie and in what areas existing systems may be improved. We conclude that most systems have a common architecture with a set of common subtasks, in particular detecting follow-up questions and finding referents for them. We characterise these tasks using the typical techniques used for performing them, and data from our corpora. We also identify a special type of follow-up question, the discourse question, which is asked when the user is trying to understand an answer, and propose some basic methods for handling it.

1 Introduction

Recent research on question answering (QA) mainly focuses on information retrieval (IR) on unstructured databases using natural-language questions. This provides an attractively non-labour-intensive way of providing user-friendly access to unstructured data. The unstructured QA community has shown increasing interest in interactive QA. The general definition of interactive QA that we use here is any means by which users can refine the research result or do a new search in the context of previous searches, so that QA effectively becomes an iterative process. In contrast with information-providing systems based on highly structured knowledge (such as regular database-oriented dialogue systems or expert systems), typical unstructured

QA systems still have no or limited dialogue capabilities. One of the problems is how to enable the QA system to understand (natural language) dialogue concepts to a useful degree without introducing knowledge- and labour-intensive techniques.

This paper addresses some of the basic research questions in this area. It continues the line of research laid out by a previous paper (van Schooten and op den Akker, 2005a), which discusses the range of basic methods a system can use to handle text-based follow-up utterances (FU), that is, follow-up questions (FQ) and other utterances, as uttered by more or less naive, non-expert users. We define an FQ as any utterance that can in some way be interpreted as a question about the subject domain, and is related to previous utterances in the dialogue.

We shall concentrate on follow-up questions (FQ) here, and present a more thorough study of these, taking advantage of experience with both the IMIX (medical domain, desktop multimodal, non-factoid, non-keyword-based) (Boves and den Os, 2005) and Ritel (open domain, telephone speech, factoid, keyword-based) (Galibert et al., 2005) QA dialogue systems, for both of which dialogue management functionality has been developed, and corpora have been analysed. We present a comparative study of alternative FQ handling techniques, covering several new areas: multimodal input and output, speech input, and the practical implementation in factoid versus non-factoid, and keyword-based versus non-keyword-based QA systems. This results in a proposal of a basic architectural framework for handling FQ in a wide range of possible situations. We shall illustrate it using data from several corpora. We shall also compare our findings with other interactive QA systems from the literature where possible.

User interface paradigm. The QA concept has already been used in serious user applications for awhile. One important emerging question is how QA functionality should be incorporated in a broader user application involving interactivity. According to classical HCI theory ((Dix et al., 2004)), we can, for example, choose between a GUI interaction paradigm or a conversational interaction paradigm (i.e. use of natural language dialogue). Since QA is already an instance of the conversational paradigm, NL dialogue is arguably a natural choice. Indeed, much interactive QA research focuses on the conversational paradigm. Follow-up question handling is then a natural extension towards including context within the NL dialogue paradigm. This is, however, a *design choice*. In fact, most real applications lean more towards the GUI interaction paradigm, presenting the top ranking answers with clickable links to the source documents. Clicking on a source document is one way for a user to pose an "FQ", analogous to asking for more context. Other GUI-base choices for FQ are imaginable, such as using a simple "search in last results" option to incorporate search context. GUI and conversational interaction can also be combined, of course. Which choices are best would require a comparative user evaluation between different finished systems, which we leave open as an interesting future direction.

In this paper we shall focus on the conversational paradigm. Generally, conversational systems are understood as having the advantage of catering for non-expert, rather than expert, users, because of their relative naturalness (i.e. analogy with human-human communication). If we had expert users, users who use QA every

day as they would a search engine, we would likely consider a different kind of user interface, giving more control but involving a more difficult learning curve. In both the IMIX and Ritel project, our basic assumption is that we are dealing with non-expert users. This implies that FQ handling in these systems should follow closely what real users do spontaneously, as is essential in NL dialogue design. This is why we shall place considerable emphasis on using corpora.

Corpora. In this paper we shall mainly discuss results from four corpora. The first two are the follow-up question corpus (van Schooten and op den Akker, 2005a), composed of 575 text FQs, and the multimodal follow-up question corpus (van Schooten and op den Akker, 2007), composed of 202 multimodal (text + gestures) FQs to multimodal (pictures+text) answers. Both are Dutch-language. These are based on presenting users with canned questions and answers, which they can respond to by uttering an FQ. The third and fourth corpus are collected from user dialogues with two versions of the Ritel system. We will call these the old Ritel corpus (Rosset and Petel, 2006), and the new Ritel corpus (van Schooten et al., 2007). These are French-language. We will also include some results from FQ dialogue corpora from the literature, in particular (Bertomeu et al., 2006) (English) and (Kato et al., 2006) (Japanese).

2 Follow-up question handling in different systems

The manner in which FQ can or should be handled depends on the features of a particular QA system. We shall characterise QA system features along three dimensions: the available modalities, the ability to answer certain types of questions, and the interface between the QA engine and the dialogue management facilities. Within this framework we shall place Imix and Ritel, as well as the corpora we collected and systems from the literature. This will form a basis for the rest of the article.

2.1 Available modalities

Most QA systems only handle text (typed) questions, answered by text answers. A minority of systems, such as Ritel, handle speech input, which is already quite a different kind of game, because of the speech processing problems of large-vocabulary ASR. Few QA systems, among which are the IMIX and SmartWeb (Reithinger et al., 2005) systems, handle multimodal FQs to multimodal answers.

Speech. ASR output for QA is typically so noisy that one can expect something like half of the output being unuseable. It is recognised that the most serious problem with the current Ritel system lie in the quality of ASR, with a word error rate of about 30%, and the error rate of keywords being around 50% (van Schooten et al., 2007). In IMIX, we don't even have serious ASR. That is, as yet the ASR only operates in a very limited subdomain. Speech requires repair dialogue, which is treated as a separate subdialogue in Ritel, and is as such independent of FQ

handling. However, speech may influence user behaviour, and may make certain FQ more or less common or desirable. For example, anaphoric FQ would especially have added value in speech QA, as they reduce the need for repeating the keywords that are so difficult to recognise by an ASR.

Multimodality. We define multimodality as the combination of the presentation of multimodal answers (text + pictures) with multimodal FQs (text/speech + pointing or gesturing on the screen). Multimodality requires an extra interpretation and/or generation step in the QA process.

While IMIX is unique in handling multimodal FQ, there are of course multimodal "question answering" systems which are built with a philosophy different from QA systems (that is, knowledge rich), for instance the Andersen system (Martin et al., 2006).

We shall assume that multimodal FQ can be handled in basically the same way as unimodal ones, except that there are specific classes of FQ that only exist in the multimodal case. In (van Schooten and op den Akker, 2007), we found that almost all multimodal FQ in our corpus were primarily linguistic with the pointing gestures being used to disambiguate the references made in the linguistic component of the utterances. This is mostly consistent with findings in the Andersen system, which found only 19% gesture-only user turns. These can be interpreted as simple "What is ...?" queries.

2.2 QA / dialogue manager interface

The second dimension, the QA-dialogue manager (QA-DM) interface, assumes that we can readily distinguish between a "QA engine" and a "dialogue manager". Here, we shall call the dialogue manager the software that determines a question's context from previous utterances. It is imaginable that this process is integrated in such a way that it cannot be practically separated, but in practice it is usually quite clear how context is calculated and passed to a QA/IR system.

In previous work (van Schooten and op den Akker, 2005a) we distinguished between "black box" and "open" QA. The difference between Imix and Ritel is a good example of a black box versus an open QA. Imix has a strict black box QA: it only enables isolated full NL questions as input. Ritel is an open QA: it enables the DM to pass any set of tagged keywords and a question type to the underlying IR engine. This interface determines the possibilities of passing context. The black box model has the least possibilities but has the advantage of modularity. The black box model requires the DM to rewrite any FQ into a self-contained question, which, as we found in the IMIX project, is both difficult and conceptually problematic (van Schooten and op den Akker, 2005a). The IMIX system does take advantage of the modularity thus obtained, in that it currently has three QA engines, based on quite different principles, which can be used simultaneously and interchangeably to search for an answer.

2.3 Ability to answer different types of questions

The third dimension involves a relatively complex issue, namely, what kind of typology do we use for questions? We shall give a broad classification, based on both QA conventions and on our experience. Generally recognised in QA are several distinct, rigidly defined, question types, in particular factoid, list, definition, and relationship questions (Voorhees, 2005). Other suggested question types are analytic (HITIQA) (Small et al., 2003), complex (a set of simpler questions, FERRET) (Hickl et al., 2006), encyclopedic (IMIX) and yes/no (IMIX). The main distinction that we will consider here is between factoid and encyclopedic. We consider encyclopedic to be similar to definition and analytic, and we shall assume that list, relationship, and yes/no question types are special cases of factoid, that is, they are basically factoid with special search criteria.

This classification concerns isolated questions only. When we look at FQ, we can observe that the conventional QA paradigm in many or most QA systems (such as those participating in TREC (Voorhees, 2005) and QAC (Kato et al., 2004)), is that an FQ is a regular QA question. That is, it can be handled by the same process that handles an isolated QA question. While older, knowledge-intensive QA systems were built to handle semantically and pragmatically different kinds of FQ, in unstructured QA, an FQ is limited to whatever you can feed into an IR engine that's basically made for isolated questions. Consequently, FQ handling methods typically concentrate on how to adapt the input to the standard QA or IR process in order to include the question's context. We will argue that the typology of FQs may be different enough to warrant a different answering process. In the next sections we explore which answering strategies are best for what kinds of FQs. The examples below illustrate the kinds of FQ that we wish to examine in this paper.

Typical TREC-style factoid follow-up question sequence:
Who is the filmmaker of the Titanic?
Who wrote the scenario?
How long did the filming take?

Possible non-factoid follow-up question sequence:
What do ribosomes produce? (factoid initial question)
All types of proteins?
And how does that work?

Some "real life" FQ from the corpora:
So the answer is "no"?
Shorter than what? Another form of schizophrenia?
How often is often?
What are these blue spots? (multimodal FQ)
Is this always true? (a reaction to an explanatory answer)
What does 'this' refer to? (reference to a pronoun in the answer)
One minute per what? (reply to a factoid answer "one minute")
What is meant here by 'hygienic handling'?

Table 1. *Overview of existing QA systems that handle FQs. The* **discourse** *column indicates that a system handles discourse questions. The* **domain** *column gives a coarse characterisation of the domain, with open being an open-domain system in the classical sense. GUI indicates that the system uses GUI-type (rather than NL-type) interaction to FQ handling (as explained in section 1). Qtype indicates that the question type (that is, person, date, number, etc) may be passed as dialogue context to the IR. Keyword or kw indicates that keywords or key phrases may be passed as dialogue context. Blackbox indicates that only full NL questions may be passed to the IR.*

System	Reference
Hitiqa	(Small et al., 2003)
De Boni et al.'s system	(De Boni and Manandhar, 2004)
SmartWeb	(Reithinger et al., 2005)
Rits-QA	(Fukumoto et al., 2004)
Nara institute's system (ASKA)	(Inui et al., 2003)
KAIST	(Oh et al., 2001)
NTU system	(Lin and Chen, 2001)
OGI school's system	(Yang et al., 2006)
FERRET	(Hickl et al., 2006)

System	language	speech	multimodal	discourse	question types	QA-DM interface	domain
Ritel	French	yes	-	-	factoid	qtype+kw	open
IMIX	Dutch	-	yes	yes	encyclop.	blackbox	medical
Hitiqa	English	-	-	GUI	analytic	GUI	news
De Boni	English	-	-	-	factoid	N/A	open
SmartWeb	German	yes	yes	GUI	factoid	GUI	multiple
Rits-QA	Japan.	-	-	-	factoid	blackbox	open
Nara	Japan.	-	-	-	factoid	qtype+kw	address
KAIST	English	-	-	-	factoid	qtype+kw	open
NTU	English	-	-	-	factoid	keyword	open
OGI	English	-	-	-	factoid	keyword	multiple
FERRET	English	-	-	GUI	complex	GUI	news

2.4 Overview of existing systems

Table 1 attempts to give an exhaustive list of QA systems handling FQ found in the literature and compares these systems according to the features we just identified. The next sections include comparisons of these systems where possible.

3 Understanding user FQs

In this section, we look into more detail at the nature of real FQ, focusing on the relationship between kinds of FQ and potential answering strategies. We will also discuss the relevance of the user's model of the system, called the *mental model* (Dix et al., 2004), and the way the system designers tries to shape it (we will call this the *designer's mental model*).

When going towards real-life QA and dialogue systems, we should consider not only the range of formally-defined question types that we support, but also the range of real questions that users ask when interacting with the system. Users will translate their information need to a question or set of questions depending on their mental model. The mental model depends on the instructions that users are given before working with the system, or feedback while working with it. It is largely an open question how mental modelling (that is, both shaping and analysing mental models) and support for different question types relate in a real-life QA dialogue system. With help of the corpora we collected and corpora from the literature, we will try to answer some of the basic questions regarding mental model.

In both IMIX and Ritel, we collected corpora of real user utterances. The users were more or less naive, and have no special knowledge of QA systems, nor did they have prior experience with these systems. We discuss the experimental method and summarise user behaviour here.

Ritel. In the Ritel dialogue corpus experiment, the users were instructed by means of a set of 300 example questions, and were not instructed to pose factoid questions, but were invited to pose any kind of question. The majority actually invented their own questions. Yet, users seemed to understand quite well what was meant, and only 12% of the questions were not factoid. Most were list and yes-no questions (7.3%), the rest were definition and other kinds of questions. The FQs users posed also neatly fell into the factoid category. Apparently, users found the factoid concept very easy to understand, though it is theoretically possible that the range of questions the users dared ask was *more* limited than desirable. We know that certain question types implemented in Ritel hardly ever seemed to occur. In our current experimental setup, we cannot know, however, if this is because of the users' mental models or because there was no demand for such questions. The experimental setup appears quite succesful in constraining the user questions to factoid questions. In contrast, (Kato et al., 2006), who expressly instructed users to pose "factoid" questions, found that their users posed some 34% non-factoid questions, mostly why, how, and definition questions.

Imix. In the IMIX FQ corpus experiments, the users were instructed to ask free-form FQ about the general medical domain (in particular excluding the important subclass of diagnostic questions, which we chose not to support). Before posing their FQ, they had already selected a number of example questions and had been shown several example FQ. The actual questions users asked were more complicated than in the Ritel case. Most user FQs were in the "encyclopedic" category, but

a significant minority of questions failed to fit into this category. We found, for example, a minority of elaborate and complex questions, some of which did fall into the diagnosis domain. But we also found that some of the questions that seemed to come naturally do not fit into the regular QA paradigm at all, and seem to require different machinery to answer. This confirms observations done as early as (Moore, 1989), who found that users often demand elaboration, with different kinds of elaboration being appropriate in different situations.

Interesting about these experiments were the level to which users would stick to particular kinds of question. We did find that users tended to copy the given examples rather than follow the instructions, but we would need more data to know for sure how to "steer" users in a particular direction.

The "real life" examples in section 2.3 illustrate some of the "exceptional questions" that we found that do not seem compatible with current QA engines. The main part of these questions is covered by what we defined as *discourse FQ*, or *discourse question* (Theune et al., 2007). These are questions that only make sense within the specific discourse of an answer. In particular, this includes questions that ask for further explanation of the answer's specifics, such as clarification of anaphors, questions about the meaning of pictures or phrases, or questions about the source or validity of the answer. Discourse questions cannot be answered except by considering the answer *itself* as the source for new questions. In fact, in our annotation schemes, we defined a discourse question as a question which cannot be made self-contained unless a significant part of the previous answer is quoted literally. We can lead discourse questions back to a special user goal, namely the goal of trying to understand a given answer. Arguably this is just a natural part of a more comprehensive QA system that is to be used by real users, naive or expert.

For structured, knowledge-intensive QA systems, such as expert systems, discourse questions are less of a problem, as the system maintains detailed dialogue knowledge anyway. However, for unstructured, knowledge-poor QA, we argue that discourse questions are a major class of FQ that are currently not handled in unstructured QA systems. Handling could be done for example by enabling the user to effectively browse through source documents, or examine and refine the process that led to the answers.

In this light, it appears to make sense to determine how much context a QA system is going to show by default. In earlier versions of Ritel, we only showed the answer, but found that this is definitely not enough, because the user cannot be sure if the ASR recognition was correct. We found that including the most important search keywords and recognised question type helped the user a lot in interpreting the result. As regards the source document context, the literature suggests that showing part of the source document by default is also a good idea, and reduces the amount of information need that users have (Lin et al., 2003). The more context we show however, the more material a user has to ask questions about.

We tried to find out how often users posed discourse questions in response to factoid answers (single-phrase answers), factoid answers with a one-sentence context (the sentence around the answer phrase is shown), and encyclopedic answers (the answer consists of one or more sentences). See table 2. We used the unimodal FQ

Table 2. *Occurrence percentages of discourse FQs in response to answers of different types in the IMIX corpora. The first three columns represent a subdivision of the correct answers in the unimodal FQ corpus, with percentages given in the second row. Factoid means just a word or phrase answer to a factoid question. Factoid with context means a factoid answer including a context sentence. Encyclopedic means the other, non-factoid answers. The fourth column represents the set of incorrect questions from this corpus. The fifth column represents the multimodal FQ corpus.*

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

corpus, in which the types "verify-question" and "missing-referent" (van Schooten and op den Akker, 2005a) were considered discourse questions. We found that the percentages of discourse questions for factoid and encyclopedic answers was similar, with factoid being slightly higher (12-13% versus 9%). We also found that discourse questions were much more often asked when the answer given to the first question was wrong. For multimodal FQs to multimodal answers, a much larger percentage of questions were discourse questions (mostly asking about the meaning of pictures). Apparently, pictures elicit a lot of clarification questions as compared to text.

Discussion. We found that mental model limits the possibilities for the types of FQ that will be asked. We also found that the designer's and user's mental models correspond reasonably in our experiments, but do not always do so in the literature. It appears that the "factoid" concept is easy to understand, if illustrated by examples. We identified a new type of user goal, answer comprehension, which we expect to play a role in any full-fledged QA system, factoid or not. In other words, with each question the user either wants one of two things:

1. *to know the answer to a specific question.* New questions and different kinds of FQ fall into this category.
2. *to comprehend a given answer.* The user is trying to interpret the content of the given answer. This is implied by a discourse question.

Apparently, the mental model in the Ritel experiments does not fit naturally with the answer comprehension user goal, as the corpus did not contain any discourse questions. Nevertheless, the goal is meaningful in the factoid context, as the subset of factoid answers in the IMIX experiments did elicit as many discourse questions as did non-factoid answers. The question remains how the mental model should be shaped to allow for them when we want to. We shall devote a separate section, section 4.2.4, on handling discourse questions.

3.1 User utterance typology

In this section we present a general classification of FQs, based on the dimensions we defined. The top-level classification is primarily based on the question type dimension: self-contained, regular FQ, and discourse question.

1. *Self-contained question.* While this is apparently a self-explanatory concept, we found that there are different species of "self-contained". In the corpora there are many questions where adding context is not required, but not harmful either. This indicates that there is a range of options here. At one end we have the *harmfulness* condition, which indicates that adding any context is harmful (these are your classical topic shifts), on the other end we have the *insufficiency* condition, where *not* adding context makes the question insufficiently complete to answer. What is in-between are basically the self-contained on-topic FQ. It is not clear which of the conditions is best for QA performance. The insufficiency condition means we will have less chance to erroneously add context when we don't really need it anyway, but knowing that a FQ is on-topic, even while self-contained may have added value for IR or dialogue management. We find, however, that different systems emphasise either of the two conditions and not the other, even though most systems do not explicitly make this distinction. De Boni (De Boni and Manandhar, 2004) for example, equates topic shifts with the distinction FQ / new question.
2. *Regular FQ, needing dialogue context information to be understood.* We distinguish three techniques for adding context, mainly based on the QA-DM interface dimension.
   (a) *Rewriting.* The FQ is rewritten to a self-contained question that can be answered by the particular QA engine.
   (b) *IR Context completion.* Context can be included by including extra search terms from the dialogue context.
   (c) *Answer document set.* The previous answer document, or document set, may be used as the source for searching for the answer.

   IMIX currently implements method (a), while Ritel currently implements method (b).

   In case the context completion consists of adding all search terms of the previous question to a regular IR based only on search term occurrence, method (b) and (c) coincide. This, plus the black box QA / open QA distinction as a basis, was reason for us to lump together method (b) and (c) in previous work (van Schooten and op den Akker, 2005a). In practice there are differences between the two, for example if we only include a selected part of the search keywords or if IR is done using mutual proximity of keywords in the documents, as is done in Ritel. In addition, we believe that method (c) may be *conceptually* different from the other two. A document can be expected to have a certain level of cohesion of information, and the success of method (c) depends on this cohesion. This of course depends on the type of document. In IMIX, we use medical encyclopedias, which can be expected to have a particularly high level of cohesion as regards medical topic (organ, disease, etc.).

Other documents, such as history books, may be coherent wrt time and date, etc. Consider the following two examples:

What is the capital of France?

Who is its mayor?

versus:

What is the length of the Mississippi?

And that of an average car?

Intuitively, the first example stays on topic, while the FQ in the second only has a tenuous link to the original question, namely the topic of heights. The second example sounds unnatural in fact, and one would expect sequences of this type to be much rarer. Conceptually, this may be linked to information need. In the first example, the user probably wants to know more about the capital of France, while in the second, it is not quite obvious what the user's information need is, if there is a coherent information need at all. The corpus study of (Bertomeu et al., 2006) indicates that both types of FQ occur in real dialogues. They found that 11% of all FQs were like the latter example, having only the answer type in common, which they attributed to users who prefer to order questions on different topics by answer type. Different systems do not seem to agree on the semantics of these two FQ. The latter FQ is considered a topic shift in Ritel (actually leading to an insufficiently completed FQ), while in De Boni, it is considered a topic shift only if the latter FQ is self-contained (while, according to the Bertomeu corpus, it may well be considered a self-contained on-topic FQ).

The significance of this is that, the more coherent the FQ, the more likely that the answer can be found in the same document where we find the original answer. The answer to the first FQ is likely to be found in a document where we find the initial question, for the second FQ that is unlikely, unless the document is structured in a special way. Arguably, the choice of IR method does not only depend on technical ability to use the method, but also on the nature of the FQ and documents. Since each method has its own advantages and disadvantages, a hybrid model might perform best. Possibly, the notion of topic shift used in existing systems could be made to relate to the type of coherence found in the source documents, and hence, be used to select the IR method.

3. *Discourse question.* The item that is referred to may be a picture or a piece of text. While thorough classification of this kind of question is an open issue, in our corpora we found a number of common discourse questions. Based on (van Schooten and op den Akker, 2005a) and (Theune et al., 2007), we arrive at the following types:

   (a) *Missing referent.* The user suspects that an antecedent of a referent did not get included in the context presented to the user.
   (b) *Verify.* User requests information to validate the answer.
   (c) *Visual element identity.* User asks for the name or identity of visual element in picture.
   (d) *Visual property.* User asks something else about the meaning of a picture.

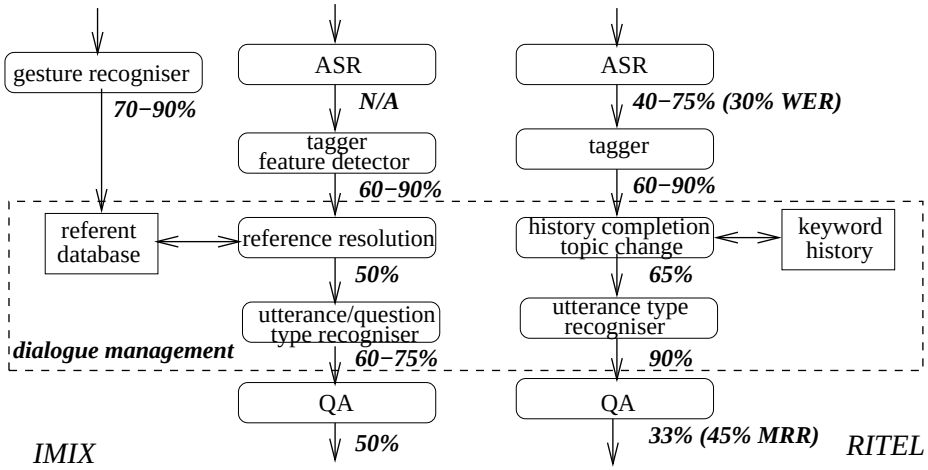

Fig. 1. Comparison of IMIX and Ritel processing pipelines. Input at the top is the user utterance, output at the bottom is system answer.

Note that this is the only part of this typology where the modality dimension plays a role.

4. *Not really an FQ, but something that looks like a question, such as negative feedback or a meta-question.* For example, "Yes, but I asked about ...", or "Is that really true?". This type of question requires alternative handling, which is discussed in (van Schooten and op den Akker, 2005b).

4 Implementation in Ritel and Imix

First, we shall provide a global overview of the Ritel and Imix architectures, and give some general information about the components and their performance. Then we shall go into more detail on the processes that have to be performed and choices that have to be made by these components, and how these impact the abilities and performance of the overall QA. Here, we shall include other systems from the literature where possible.

4.1 comparison of architectures

We will compare the IMIX and Ritel architectures (figure 1) alongside each other, and give an indication of where bottlenecks are and where errors occur in the processing pipelines. We shall give a summary of the different processing stages and give approximate performance figures.

Approximate performance:

- *ASR.* Ritel uses a large-vocabulary ASR (about 65,000 words total). The Ritel ASR has a word error rate of approximately 28-30%. In the Ritel corpus we found that for 24% of the utterances the ASR result was useless, 38% were partially useful, and only 38% were basically correctly recognised. In IMIX, we do not have an ASR with a sufficiently large domain.

As regards confidence, neither ASR generates a proper confidence value. We tried determining confidence in Ritel by means of postfiltering the ASR output, but failed to find an effective postfiltering method.

- *Tagger and feature detector.* The Ritel tagger tags a mix of semantic tags and POS tags, for example: person, location, event, time, NP, PP. There are about 250 tags in all. The performance is 80-90% for named entities, and mixed for other tags.
  IMIX combines a tagger and a feature detector in this stage, tagging POS tags, semantic tags, a dependency tree, and a set of features such as: "the sentence is elliptic", "the sentence has a question form", etc. IMIX semantic tags are for example: disease, person, treatment, and relations like causes, treats (about 15 in all). The semantic and POS tags have a performance of about 80-90% or more, the other tags have a varying performance.
  Only a minority of the taggers include confidence values. Like the ASR stage, we have a shortage of confidence information. The performance is pretty good however.
- *Reference resolution and history completion.* The IMIX reference resolution has an overall performance of 51% for the rewritable questions in the unimodal FQ corpus, and approximately 49% on the appropriate questions in the multimodal FQ corpus. The Ritel history completion algorithm yields meaningful completions for about 65% of the FQ.
  There is a difference in the two modules, namely, Ritel first determines whether history completion is meaningful, and, if so, completes it, while IMIX comes up with potential reference resolutions, and determines whether reference resolution is meaningful in the next stage in the pipeline. In Ritel, we found that the topic detection task in particular often comes up with false positives, explaining part of the errors. In IMIX, we find that both detection of rewritability and antecedent selection are error-prone.
- *Utterance and question type recogniser.* The performance figures here assume that the user poses an FQ. IMIX decides at this point whether to use the resolved references found in the previous stage, and how. Both systems determine at this point whether the utterance is a question or not. The IMIX dialogue act recogniser has a performance of 60-75% (as tested on the follow-up question corpus), the Ritel recogniser only has to distinguish between questions and other types of utterances, such as negative feedback, and has a performance of about 90% for correctly recognising questions. Ritel outputs no confidence information here, but IMIX outputs a confidence value which is used to determine if the system should ask the user for disambiguation or clarification.
- *QA engine.* Ritel's QA module involves question type recognition and the actual IR. As tested with CLEF 2005, Ritel performed with a mean reciprocal rank (MRR) of 45%, and 33% of the top answers were correct. The IMIX QA involves three different QAs running in parallel. One of these has been tested in the open domain, with a performance of 50% correct answers as tested with

CLEF 2005. Both systems output confidence values for the given answer, but the reliability of these is questionable.

Discussion. We showed that the architectures of the two systems, and the performance of the different processing stages, are comparable. The main bottlenecks are: ASR (which is dealt with using confirmation, which is out of this paper's scope), reference resolution (which will be discussed here), FQ type recognition (which will be discussed here in part), and QA (which is out of this paper's scope). At the places where confidence information is missing, which is often, we generally employ user feedback to make decisions.

4.2 Comparison of context completion

Context completion, what happens inside the reference resolution and history completion components, follows the following basic strategy. We shall give performance figures for the different systems where available.

1. *Identification of need for context completion.* This is analogous to detecting if the question is self-contained or not. This is the most basic step in any QA that wishes to handle FQ. Some QAs only implement this step, then apply some very basic IR algorithm, with good results. Note that existing TREC/QAC context tracks do not address this step at all, since the TREC/QAC dialogues do not contain topic shifts.
   In case a system has support for discourse questions, they can be detected in this step, as IMIX does.
   Performance is measured by the percentage of correct (neither harmful nor insufficient) classifications. Performance baseline is choosing the most often occurring class (which is typically that context is needed).
2. *Identification of rewriting strategy and anaphors.* In case we are trying to rewrite the question into a self-contained question, we need to find out how it should be rewritten. Some systems that do not require rewriting to obtain proper input for the IR may still require some structural properties of the question to be passed to the IR, in which case this step must be performed partially. If we are not rewriting the question but just passing "bag of words" information directly to the IR engine, as we do in Ritel, we may skip this step entirely.
   This step is explained in detail in (van Schooten and op den Akker, 2005a), so we will devote less attention to it here. Previously, we suggested choosing one out of a small set of relatively basic rewriting strategies. We found that the most successful strategy by far for both unimodal and multimodal FQ has been the anaphor substitution strategy. Here, an anaphor in the sentence has to be located and identified as being substitutable. In practice, we found that only a fraction of typical FQs can be rewritten at all, due to the lack of proper rewriting methods to cover the entire range of FQs satisfactorily.
   Performance is measured by assuming that the first step was done correctly,

and by counting the percentage of correct (intelligible and syntactically correct) sentences. Any simple baseline is likely to come up with very low performance, so we arguably do not need a baseline to compare with.

3. *Referent selection.* Of the three strategies identified in section 3.1, we can say that both rewriting and IR context completion require referents from previous utterances to be identified and selected, while the answer document set strategy doesn't. Each question and answer is scanned for potential referents which are stored in the referent database. For multimodal answers, the referents include the pictures and the visual elements or text labels within the pictures. Unlike the other referents in the database, these are retained only as long as the answer remains on screen. Referent selection then amounts to the selection of suitable referents from referents previously entered into the referent database.
   For multimodal utterances, picture annotations and gesture recognition is necessary as well to perform this step, see section 4.2.3.
   Performance is measured by assuming that previous steps were performed correctly, and counting the number of cases that no harmful antecedents neither insufficient antecedents were selected. We argue that a baseline is applicable here. Selecting the most important keywords (such as all named entities) from the previous question has a relatively high chance of success. In fact, the OGI system uses this baseline method with success (Yang et al., 2006). Additional evidence is that the IMIX follow-up question corpus also shows that 75% of the anaphoric references refer to a domain-specific noun phrase in the original question. (Bertomeu et al., 2006) also found that 53% of all FQs in their dialogues refer to a topic introduced in the previous question.

In table 3 we summarise existing systems in terms of these steps, and give performance figures where available. We have to conclude that the performances are hard to compare, not only because the systems have different languages and domains, but also use different performance criteria and corpora. We find that the same system tested on different corpora can give quite different results. We also find that no distinction was made between harmful and insufficient anywhere.

In the next sections we will discuss implementation details of each of these steps, and give some performance figures and conclusions.

4.2.1 Identification of need for context completion (step 1)

This step receives significant attention in the literature, so it makes sense to devote a separate section to it. In this section we discuss the algorithms and features used to detect the need for context completion.

We will first look at the distinction between self-contained and regular FQ. The philosophy of the Ritel and IMIX systems are markedly different here. The difference may be related to the harmful vs insufficient conditions. In fact, some of the problems found may be better understood with help of these concepts.

Ritel uses the notion of *topic shift* which is meant to indicate that it's harmful to

Table 3. *Context completion steps performed by different systems, and some performance figures. "Yes" means the step is performed but no figures are known; "b" means baseline performance.*

system	need-context	rewriting	ref-select	overall
Ritel	yes	-	yes	
IMIX	75%(b=61%)[1]	yes	yes	14%[2]
De Boni	83%(b=62%)[3]; 96%(b=78%)[4]	-	-	N/A
Rits-QA	-	yes	yes	37%[6]
Nara	-	-	yes	100%[7]
KAIST	-	yes	yes	[8]
NTU	-	-	yes	[8]
OGI	93%(b=62%)[3]; 74%(b=64%)[5]	-	yes	84%[9]

(1) - unimodal FQ corpus, classification includes discourse question
(2) - unimodal FQ corpus, rewriting and ref-select combined
(3) - sequence of TREC-10 context dialogues
(4) - De Boni dialogue corpus
(5) - HANDQA dialogue corpus
(6) - QAC2 corpus, overall rewriting performance
(7) - overall context completion performance using restricted language dialogue
(8) - TREC-10 context task participants, no results
(9) - retrieval performance in top 50 documents, TREC-10 and TREC 2004

use the context completion machinery when the user has changed to a completely different topic. Ritel sets a context completion prohibition flag if topic change is detected. Topic shift is also used by the De Boni and OGI systems. In fact, these two algorithms are based on detecting the boundaries of concatenated sequences of unrelated dialogues or TREC FQ series.

IMIX, on the other hand, tries to make a distinction between questions on the same topic that do and do not require context, even if they are follow-up questions. IMIX is trying to be as lazy as possible as regards context completion, because its particular algorithm is relatively error-prone. It is primarily based on distinguising between different kinds of FQ in the FQ corpora, in which there are no topic shifts. The paradigm used here is basically the insufficiency condition.

When we look at the TREC and QAC corpora used in the experiments listed in table 3, we find that *all* FQs require context, as we might expect for a context completion test. The concatenation of sequences of these FQs will lead to FQs which are either topic shifts or require context, and will have no FQ that are in-between harmful and insufficient. This contrasts with real user dialogues. We found that 25% of the FQs in the unimodal FQ corpus, and 18% of the FQs in the multimodal FQ corpus were in-between, so it makes sense to account for these. (Kato et al., 2006) also reports 26% self-contained FQs in their dialogue corpus. In the new Ritel

corpus, we even found that 63% of obviously on-topic FQ were self-contained (for some reason, users tended to remain unusually self-contained in this setting).

If we look at the features that different systems use for this step, we may distinguish the following major classes:

1. *Presence of pronouns and anaphors.*
2. *Ellipsis.* Ellipsis is typically defined as the absence of a verb.
3. *General cue words.* Examples are "so", "not", "but", etc.
4. *Presence of keywords specific enough for successful IR.* This involves seeing if there are enough specific keywords in the sentence for it to possibly be a self-contained question. If none are found for example, the question is not likely to be self-contained. We define a keyword as specific when it's worth asking a question about on its own. We assign the specific label to a keyword X when the question "What is X?" would be a sensible QA question for the typical user.
5. *Semantic distance of keywords with those of previous utterances.* If there is a semantic closeness between keywords of two utterances, we expect them to be on the same topic. Semantic closeness can be calculated using distance in terms of ontological relations, and by detecting semantic interrelations between different keywords.

For the systems that perform this step, the following features are used:

system	pronoun	ellipsis	cue words	keywords	distance
Ritel	-	-	-	-	yes
IMIX	yes	yes	yes	yes	-
De Boni system	yes	yes	yes	-	yes
OGI system	yes	yes	yes	-	yes

The most natural feature for topic shift detection appears to be semantic distance. In Ritel, this is the only feature used. De Boni and OGI use it too, and combine it with pronouns, ellipsis, and cue words. However, semantic distance has a weakness we already identified: it can never detect the difference between a new question or an FQ on the same topic. De Boni et al. noted this as a problem, and Yang et al. also found that their algorithm performed much worse when they applied it on a new, narrow-domain, corpus. One of the reasons was that the typical semantic distance between the different questions in the domain was too small. IMIX primarily tests for insufficiency, and uses all of the above features except semantic distance. Instead, another feature, number of relevant keywords, is used, which is more in line with the insufficiency condition.

As regards performance impact of these different features, we can say that all of them have some added value in improving performance, though it is at this point difficult to compare the relative expected value. What we can say is that semantic distance is by far the feature that is hardest to implement, as it requires a good semantic model. De Boni notes problems with the Wordnet model used, and in IMIX, the lack of an appropriate ontology was the main reason not to use it.

Discourse questions. The distinction between discourse question and regular question is a second problem to be tackled. IMIX does this by mainly looking at specific cue words and sentence patterns. There is one interesting feature that can be specifically used for this problem, namely whether the question refers to the previous answer or not. While a question referring to something in a previous answer does not mean that we're dealing with a discourse question, we are *not* dealing with one when it does *not*. The corpus analysis in (Bertomeu et al., 2006) also indicates that this may be related to syntactic form. For example, they found that deictic NPs and anaphors are more often correlated with an FQ that refers to a new item introduced in the previous answer than other syntactic forms.

Discussion. Topic shift detection is the main paradigm used for identifying the need for context completion. However, it is theoretically flawed, or rather, incomplete. The ability to detect it remains very useful, if complemented with other techniques. Since we can expect some 20% of FQ to be on-topic, yet self-contained, we propose that more emphasis should be laid on the insufficiency condition. Detecting that an on-topic FQ is self-contained may help improve IR.

Some of the features used here may also have relevance for the context completion technique that should be used, although no systems implement this at the moment. A non-self-contained FQ with topic shift, for example, is a cue for *not* using the "answer document set" technique, as we indicated in section 3.1. Following (Bertomeu et al., 2006), we might also use some of the other features to detect this case. That means it may be meaningful to perform a four-class classification in this step, namely: self-contained question, on-topic FQ, FQ with topic shift, and discourse question.

4.2.2 Antecedent or keyword selection and formulation (step 3)

The main task in this step is selecting appropriate referents from the database. This is done by ranking the antecedents and matching them with the FQ. We distinguish five criteria, based on both traditional anaphora resolution and techniques found in QA dialogue systems, thus providing a full coverage of existing features.

1. *Semantic matching.* Checking if the query is semantically compatible with certain referent keywords. Ritel implements this by finding keywords that are likely to be required to match those already present. For example, if we ask for a mayor it's likely we need a city. IMIX does not implement semantic matching.
2. *General importance of keywords.* IMIX uses the "specific" tag explained in the previous section to select porential antecedents, and in Ritel, keywords are ranked based on their type labels and position in the sentence.
3. *Salience.* Both IMIX and Ritel use a basic Lappin and Leass (Lappin and Leass, 1994) scheme. Salience of a referent decreases over time, and increases with frequency of mention.
4. *Anaphor-antecedent matching.* A classical method of selecting a referent is by matching surface characteristics of the anaphor with the referent's antecedent,

such as gender, number, or identification. IMIX implements this, while Ritel does not implement this, as it would require extra tags to be added first. The multimodal case is similar, but adds to this visual characteristics of visual elements (shape, colour, etc).

5. *Confidence.* IMIX calculates confidence according to the features used to detect the keyword. In both systems, confidence is also calculated using confirmation feedback.
6. *Deictic gestures.* This is only applicable to the multimodal case. IMIX uses gestures as strong cues for referent selection.

These features are used by the following systems (note that "?" means unknown):

system	semantic	kw rank	salience	anaphor	confidence	deictic
Ritel	-	yes	yes	yes	yes	-
IMIX	yes	yes	yes		yes	yes
Rits-QA	?	?	?	?	?	-
Nara	yes	yes	yes	-	-	-
KAIST	?	?	?	yes	?	-
NTU	-	yes	yes	-	yes	-
OGI	-	-	-	-	-	-

This step, even while complex, is not always described in detail in the literature. Different systems all use different subsets of features. As regards relative performance of the features, again we can say that most features obviously have added value (as they are after all well established in other areas of NLP), although it is not possible to make out which ones are best in general. We can again say that semantic matching is the most difficult. Also, our own experience is that anaphor-antecedent matching is error-prone (as anaphors and antecedents do not always match nicely in gender and number, for example) and is probably not useful unless well implemented.

So, we suggest that a more elaborate examination of this step using detailed performance figures is required. We also suggest a close examination of possible baselines. We suggested using "most important keywords from previous questions" to be a good baseline, as is observed in the OGI and IMIX systems and the Bertomeu corpus.

4.2.3 Additional resources needed for multimodal reference resolution

Besides the tasks mentioned, multimodal handling requires two additional tasks:

1. *Picture annotation.* The only aspect of our current FQ scheme that breaks completely with the unstructured-QA philosophy of handling data in a non labour intensive way is the manual annotation of pictures. Note that this manual annotation is only necessary for FQs, and, although it may be quite beneficial for image retrieval, it is currently not used by the IR system. So, we would like to include a note on how to get around this weakness.
   In our annotation process, we tried to leave open some possibilities for automation, and we will show how we may arrive at such data without manual

annotation. We started off with a collection of raw bitmap images, and produced the annotated pictures in two steps, first producing a set of contours from the bitmaps, then annotating the contours with semantics.
Contourisation can conceivably be further automated by using smart image filters. Both contourisation and labelling may be automated in one go by using visual processing and image retrieval techniques, which, however, would probably stretch the state of the art to the limit. Labelling may be automated by a diagram interpretation technique such as is proposed in (Futrelle and Rumshisky, 2001). Alternatively, we may step in at a different level, for example we might have access to vector graphics (in the form of vectorised medical illustrations in IMIX, for example), or even annotated vector graphics in the form of medical illustrations with callouts added digitally. Yet another option is to collect a manually annotated database using a "community annotation" technique, such as (Russell et al., 2005).

2. *Gesture referent recognition.* The interpretation of gestures is done in a separate component. This involves segmenting mouse strokes into gestures and determining what they point at. This appears to be a relatively easy problem. We found that a simple gesture system already resolves 88% of the appropriate referents, and there is potential for figures close to 100% (Willems et al., 2005).

4.2.4 Discourse question handling

IMIX handles this type of question, while Ritel doesn't. In IMIX we implemented handling methods for the four specific cases we identified in section 3.1. More detail can be found in (van Schooten and op den Akker, 2005a) and (Theune et al., 2007).

- *Missing referent and verify.* Both are handled by showing the answer's source document (which is about 1-3 paragraphs of text). In the unimodal FQ corpus, we found that this method can answer 73% of the relevant questions.
- *Visual element identity.* Handled by showing the element's annotated name. In the multimodal FQ corpus, this method can answer 84% of the relevant questions asking for identity of a visual element.
- *Visual element property.* We suggested handling these questions by showing the image caption, or other text associated with the image, when available. However, in the experimental setup of the multimodal FQ, no or little extra caption material was available, and no performance figures could be obtained.

This is only a very basic handling scheme, and we can offer some other suggestions for handling other kinds of discourse questions:

- Give other answer candidates that were returned by the IR.
- Complete the answer by including selective context from the source document, for example using RST (Rethorical Structure Theory) annotation. This does require the source documents to be annotated with structural annotations.

- Use the user feedback to disambiguate reference resolution in the source document, in case we are using reference resolution to answer encyclopedic questions.
- Answer questions about pictures by means of more extensive picture annotations, in particular annotations describing picture semantics.

5 Conclusion

We find that there is a broad range of ways in which users can express information need during a QA session. In this paper we indicated some of the possible dimensions of this issue, including the mental model, and the different types of FQ that users ask. We also tried to give form to several interesting research directions in FQ handling, in particular discourse questions and multimodal FQs.

Although existing QA dialogue systems have different approaches to FQ handling and operate in different domains, we have shown that they have certain tasks and a certain general architecture in common. We have indicated that performance evaluation depends heavily on the specific corpora used, and there is too little agreement on how to measure performance. We propose that it is possible to produce a new set of standardised evaluations of certain subtasks of FQ handling, for example in the form of competition tracks. In particular we identified the following tasks: identification of the need for context completion, question rewriting, and referent selection. We've tried to give an account of the general issues to consider in these tasks.

We proposed some basic methods for recognising and handling the user's need for comprehending answers, in the form of discourse questions. It is possible that very different approaches, such as the more traditional GUI-like systems mentioned in this paper, may address some of these goals in a better way. Possibly we can evaluate this if we can make a comparison of systems with different user interface paradigms.

Multimodal FQs are still a new research area, and we have shown how they can be handled using annotated pictures. The implementation remains infeasible, however, as long as we rely on labour intensive picture annotation techniques.

6 Acknowledgement

This work was partially funded by by the Netherlands Organisation for Scientific Research (NWO) IMIX programme, the European Commission under the FP6 Integrated Project IP 506909 CHIL, and the LIMSI AI/ASP Ritel grant.

References

Bertomeu, N., Uszkoreit, H., Frank, A., Krieger, H.-U., and Jörg, B. (2006). Contextual phenomena and thematic relations in database qa dialogues: results from a wizard-of-oz experiment. In *Workshop on Interactive Question Answering, HLT-NAACL 06*.

Boves, L. and den Os, E. (2005). Interactivity and multimodality in the imix demonstrator. In *Multimedia and Expo, ICME 2005*, pages 1578–1581.

De Boni, M. and Manandhar, S. (2004). Implementing clarification dialogues in open domain question answering. *Journal of Natural Language Engineering*.

Dix, J., Finlay, J., Abowd, G., and Beale, R. (2004). *Human-Computer Interaction (3rd edition)*. Harlow: Pearson/Prentice Hall.

Fukumoto, J., Niwa, T., Itoigawa, M., and Matsuda, M. (2004). RitsQA: List answer detection and context task with ellipses handling. In *Working notes of the Fourth NTCIR Workshop Meeting*, pages 310–314.

Futrelle, R. P. and Rumshisky, A. (2001). Discourse structure of text-graphics documents. In *1st International Symposium on Smart Graphics*, pages 31–38.

Galibert, O., Illouz, G., and Rosset, S. (2005). Ritel: an open-domain, human-computer dialog system. In *Interspeech 2005*, pages 909–912.

Hickl, A., Wang, P., Lehmann, J., and Harabagiu, S. M. (2006). FERRET: Interactive question-answering for real-world environments. In *ACL 2006*.

Inui, K., Yamashita, A., and Matsumoto, Y. (2003). Dialogue management for language-based information seeking. In *Proc. First International Workshop on Language Understanding and Agents for Real World Interaction*, pages 32–38.

Kato, T., Fukumoto, J., and Masui, F. (2004). Question answering challenge for information access dialogue – overview of NTCIR4 QAC2 subtask 3. In *Working notes of the Fourth NTCIR Workshop Meeting*.

Kato, T., Fukumoto, J., Masui, F., and Kando, N. (2006). Woz simulation of interactive question answering. In *Workshop on Interactive Question Answering, HLT-NAACL 06*.

Lappin, S. and Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Lin, C.-J. and Chen, H.-H. (2001). Description of NTU system at TREC-10 QA track. In *TREC 10*.

Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., and Karger, D. R. (2003). What makes a good answer? the role of context in question answering. In *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT-2003)*.

Martin, J.-C., Buisine, S., Pitel, G., and Bernsen, N. O. (2006). Fusion of children's speech and 2D gestures when conversing with 3D characters. *Special issue on multimodal interfaces of the Signal Processing journal*, 86(12):3596–3624.

Moore, J. D. (1989). Responding to 'Huh?': Answering vaguely-articulated follow-up questions. In *Proceedings of the Conference on Human Factors in Computing Systems*.

Oh, J.-H., Lee, K.-S., Chang, D.-S., Seo, C. W., and Choi, K.-S. (2001). Trec-10 experiments at kaist: Batch filtering and question answering. In *TREC*.

Reithinger, N., Bergweiler, S., Engel, R., Herzog, G., Pfleger, N., Romanelli, M., and Sonntag, D. (2005). A look under the hood: design and development of the first smartweb system demonstrator. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, pages 159–166, New York, NY, USA. ACM Press.

Rosset, S. and Petel, S. (2006). The Ritel corpus - an annotated human-machine open-domain question answering spoken dialog corpus. In *International Conference on Language Resources and Evaluation*.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2005). LabelMe: a database and web-based tool for image annotation. Technical report, MIT. MIT AI Lab Memo AIM-2005-025.

Small, S., Liu, T., Shimizu, N., and Strzalkowski, T. (2003). HITIQA: an interactive question answering system: A preliminary report. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*.

Theune, M., Krahmer, E., van Schooten, B., op den Akker, R., van Hooijdonk, C., Marsi, E., Bosma, W., Hofs, D., and Nijholt, A. (2007). Questions, pictures, answers: Introducing pictures in question-answering systems. In *Tenth international symposium on social communication*, pages 469–474, Universidad de Oriente Santiago de Cuba.

van Schooten, B. and op den Akker, R. (2005a). Follow-up proceedings of ntcir-5 workshop meeting, december 6-9, 2005, tokyo, japanutterances in QA dialogue. *Traitement Automatique des Langues*, 46(3).

van Schooten, B. and op den Akker, R. (2007). Multimodal follow-up questions to multimodal answers in a QA system. In *Tenth international symposium on social communication*, pages 469–474, Universidad de Oriente Santiago de Cuba.

van Schooten, B., Rosset, S., Galibert, O., Max, A., op den Akker, R., and Illouz, G. (2007). Handling speech input in the Ritel QA dialogue system. In *Interspeech 2007*.

van Schooten, B. W. and op den Akker, R. (2005b). Follow-up utterances in QA dialogue. *Traitement Automatique des Langues*, 46(3).

Voorhees, E. (2005). Overview of the TREC 2005 question answering track. Technical report, NIST.

Willems, D. J. M., Rossignol, S. Y. P., and Vuurpijl, L. G. (2005). Features for mode detection in natural online pen input. In *BIGS 2005: Proceedings of the 12th Biennial Conference of the International Graphonomics Society*, pages 113–117.

Yang, F., Feng, J., and Di Fabbrizio, G. (2006). A data driven approach to relevancy recognition for contextual question answering. In *Workshop on Interactive Question Answering, HLT-NAACL 06*.