FINAL DRAFT. Appeared in: *Logica Yearbook 2012*, eds. V. Punčochář and P. Švarný, College Publications, 2013, p. 79–90.

Cooperative question-responses and question dependency

Paweł Łupkowski*

Abstract

The concept of cooperative question-responses as an extension of cooperative interfaces for databases and information systems is proposed. A procedure to generate question-responses based on question dependency and erotetic search scenarios is presented.
Keywords: questions, dependent questions, databases, cooperative answering, inferential erotetic logic, erotetic search scenarios.

Introduction

The main aim of this paper is to propose an extension of cooperative answering techniques with question-responses. In order to obtain such responses I will use the concept of question dependency and erotetic search scenarios—a tool developed within A. Wiśniewski's Inferential Erotetic Logic (IEL).

In the first section I describe the idea of cooperative answering. I also propose to extend standard cooperative answering techniques with a capability of replying with a question. I point out motivations for this step based on the natural language dialogues. In the second section necessary concepts taken from IEL are introduced along with the idea of dependency of questions. Third section contains a description of the procedure which allows to generate cooperative question-replies. The paper ends with indicating potential advantages of supplementing cooperative interfaces with questioning capabilities.

1 Cooperative Answers or Cooperative Responses?

The idea behind cooperative answering (in the context of databases and information systems) is to provide a user with an answer to his/her query which is not only correct, but also non-misleading and useful (cf. Gaasterland et al., 1992). Let us consider a well known example (cf. Gal, 1988, p. 2) which shall shed some light on what counts as a

*I would like to give my thanks to D. Leszczyńska-Jasion, M. Urbański and A. Wiśniewski for helpful feedback and comments on a draft of this article. This work was supported by funds of the National Science Council, Poland (DEC-2012/04/A/HS1/00715).

cooperative answer. Imagine that a student wants to evaluate a course before registering in it. He asks the following question:
Q: *How many students failed course number CS400 last semester?*
Assume also that the course CS400 was not given last semester. For most database interface systems the answer to the student's question would be:
A_1^*: *None.*
This answer is correct according to the database information state. But on the other hand, it is easy to notice that it is also misleading for the student (who is not aware of the fact that the course was not given in the last semester) and thus uncooperative from our perspective. However, when we think about a secretary answering the same question we may imagine that the secretary would easily recognise the student's false assumption and correct it in her answer:
A_1: *None, but the reason for this is that course number CS400 was not offered last semester.*
The answer A_1 is not only correct, but it is also non-misleading and useful for the student. The cooperative answer given by the secretary facilitates student's further search. We may, for example imagine that the next question—asked on the basis A_1—would be: *When was CS400 offered and how many students failed it then?*

A review of the literature reveals that techniques developed in the field of cooperative answering are focused on declarative sentences as reactions to the user's queries. In fact, many authors write simply about answers (as declarative sentences). As such, cooperative answering is well explored and a number of techniques have been developed in this area of research, such as: evaluation of presuppositions of a query; detection and correction of misconceptions in a query (other than a false presupposition), formulation of intensional answers or generalisation of queries and of responses. A detailed description of the above techniques may be found in [Gaasterland et al., 1992] and [Godfrey, 1997].

In this paper I will use the distinction—introduced by Webber [1985]—between a *direct answer* and a *cooperative response* to a user's query. A direct answer is described as the information directly requested by the user's query. As for cooperative response, it is described as an informative *reaction* to the user's query (understood here as a declarative sentence or a question). Consequently we may say that the very idea of cooperative interfaces is to provide a user with a cooperative response. The introduced distinction allows us to consider many forms of reactions as responses to the user's query, however two of such reactions are most natural in the context of cooperative answering systems: declarative sentences and questions.

Benamara and Saint-Dizier [2003] gathered a corpus elaborated from Frequently Asked Questions sections of various web services. Besides well formed questions the corpus revealed some interesting types of questions like:

- questions including fuzzy terms (like *a cheap country cottage close to the seaside in Cote d'Azur*);
- incomplete questions (like *What are flights to Toulouse?*);
- questions based on series of examples (like *I am looking for country cottages in mountain similar to Mr. Dupond cottage*).

These questions were not taken into account in WEBCOOP development process (because of early stage of the project at the moment). However, this type of questions asked by users suggests that questions should also be allowed as responses in such systems. Question posing ability would enable to ask a user for missing information not expressed in his/her question in a natural dialogue manner. The motivation comes from everyday natural language dialogues. As Ginzburg points out:

> Any inspection of corpora, nonetheless, reveals the undiscussed fact that many queries are responded with a query. A large proportion of these are clarification requests (...) But in addition to these, there are query responses whose content directly addresses the question posed (...) [Ginzburg, 2010, p. 122]

This fact was also noticed by researchers working with databases. In [Motro, 1994, p. 444] we read:

> When presented with questions, the responses of humans often go beyond simple, direct answers. For example, *a person asked a question may prefer to answer a related question*, or this person may provide additional information that justifies or explains the answer. (emph. by P.Ł.)

2 Question Dependency and IEL

As recent corpus study shows, one of the most common question-responses in the natural language conversations are dependent questions (cf. Łupkowski and Ginzburg, 2013). The rationale behind dependent questions might be summarised as follows [Ginzburg, 2010, p. 123]: question Q_1 depends on question Q_2 if discussion of Q_2 will necessarily bring about the provision of information about Q_1. This allows to say that Q_2 might be used to answer Q_1—in other words Q_2 is an acceptable response to Q_1.

The following example illustrates this idea:

> A: Any other questions?
> B: Are you accepting questions on the statement of faith at this point? [F85, 70–71][1]
> (i.e. *Whether more questions exist depends on whether you are accepting questions on the statement of faith at this point.*)

One of the ways in which dependent questions might be modelled is by the use of erotetic implication (e-implication), one of the key concepts of IEL—(cf. Łupkowski, 2012).

In what follows I will use the formal language L; this language resembles a language characterised in [Wiśniewski, 2001, pp. 20–21]. The 'declarative' part of L is a first-order language with identity and individual constants, but without function symbols. A *sentence* is a declarative well-formed formula (d-wff for short) with no occurrence of a free variable. The vocabulary of the 'erotetic' part of L consists of the signs: ?, {, }, and the comma.

Questions of L are expressions of the following form:

$$?\{A_1, A_2, \ldots, A_n\}$$

where $n > 1$ and $A_1, A_2, \ldots, A_n$ are nonequiform, that is, pairwise syntactically distinct, d-wffs of L. If $?\{A_1, A_2, \ldots, A_n\}$ is a question, then each of the d-wffs $A_1, A_2, \ldots, A_n$ is a *direct answer* to the question.

[1] This notation indicates the British National Corpus file (F85) together with the sentence numbers (70–71).

A question $?\{A_1, A_2, \ldots, A_n\}$ can be read, 'Is it the case that A_1, or is it the case that $A_2, \ldots$, or is it the case that A_n?'.

Definition 1 *[Wiśniewski, 1995] A question Q* implies *a question Q^** on the basis of *a set of d-wffs X (in symbols: $\mathsf{Im}(Q, X, Q^*)$) iff*

1. *for each direct answer A to the question Q: $X \cup \{A\}$ entails the disjunction of all the direct answers to the question Q^*, and*
2. *for each direct answer B to the question Q^* there exists a non-empty proper subset Y of the set of direct answers to the question Q such that $X \cup \{B\}$ entails the disjunction of all the elements of Y.*

If $X = \emptyset$, then we say that Q implies Q^* and we write $\mathsf{Im}(Q, Q^*)$.

The first condition requires that if the implying question is sound[2] and all the declarative premises are true, then the implied question is sound as well[3]. The second condition requires that each answer to the implied question is potentially useful, on the basis of declarative premises, for finding an answer to the implying question. To put it informally: each answer to the implied question Q^*, on the basis of X, narrows down the set of plausible answers to the implying question Q.

Erotetic search scenarios may be defined as sets of the so-called erotetic derivations [Wiśniewski, 2003] or, in a more straightforward way, as finite trees [Wiśniewski, 2010, pp. 27–29], see also [Urbański and Łupkowski, 2010]:

Definition 2 *An* e-scenario for a question Q relative to a set of d-wffs X *is a finite tree Φ such that:*

1. *the nodes of Φ are (occurrences of) questions and d-wffs; they are called e-nodes and d-nodes, respectively;*
2. *Q is the root of Φ;*
3. *each leaf of Φ is a direct answer to Q;*
4. *$\boldsymbol{d}Q \cap X = \emptyset$;*
5. *each d-node of Φ:*
   (a) *is an element of X, or*
   (b) *is a direct answer to the e-node Q^* which immediately precedes in Φ the d-node considered (where $Q^* \neq Q$), or*
   (c) *is entailed by (a set of) d-nodes which precede the d-node in Φ;*
6. *for each e-node Q^* of Φ different from the root Q:*
   (a) *$\boldsymbol{d}Q^* \neq \boldsymbol{d}Q$ and*
   (b) *$\mathsf{Im}(Q^{**}, Q^*)$ for some e-node Q^{**} of Φ which precedes Q^* in Φ, or*
   (c) *$\mathsf{Im}(Q^{**}, \{A_1, ..., A_n\}, Q^*)$ for some e-node Q^{**} and some d-nodes $A_1, ..., A_n$ of Φ that precede Q^* in Φ;*

[2] A question Q is *sound* iff it has a true direct answer (with respect to the underlying semantics).

[3] This property may be conceived as an analogue to the truth-preservation property of deductive schemes of inference.

Table 1: Example of deductive database

EDB	IDB	IC
$usr(a)$ $usr(b)$ $usr(c)$ $live(a,p)$ $live(b,zg)$ $live(c,p)$	$locusr(x) \to usr(x)$ $locusr(x) \to live(x,p)$ $usr(x) \wedge live(x,p) \to locusr(x)$	$\neg(\exists x(live(x,zg) \wedge live(x,p)))$

7. *each d-node has at most one immediate successor;*

8. *Φ involves at least one e-node different from the root Q;*

9. *an immediate successor of an e-node different from the root Q is either a direct answer to the e-node, or exactly one e-node;*

10. *if the immediate successor of an e-node Q^* is not an e-node, then each direct answer to Q^* is an immediate successor of Q^*.*

The pragmatic intuition behind the e-scenario is that it '... provides us with conditional instructions which tell us what questions should be asked and when they should be asked. Moreover, an e-scenario shows where to go if such-and-such a direct answer to a query appears to be acceptable and goes so with respect to any direct answer to each query.' [Wiśniewski, 2003, p. 422]. A certain exemplary e-scenario is presented in Figure 1.

For the proposed approach the idea of a query of an e-scenario is also important.

Definition 3 *A* query *of an e-scenario Φ is an e-node Q^* of Φ different from the root of Φ and such that the immediate successors of Q^* are the direct answers to Q^*.*

As it might be noticed e-scenarios are constructed in such a way that all queries are closely related to the initial question by the dependency relation (in IEL expressed in terms of e-implication). We will use this feature in order to generate question-responses. This will ensure that question-responses generated on the basis of e-scenarios will be relevant to the user's question.

3 E-scenarios in Generating Question-responses

In this section I will introduce a simple technique of generating cooperative question-responses on the basis of question dependency check in an e-scenario. Let us assume that we are dealing with a deductive database. It consists of an extensional database (EDB)—built out of facts, intensional database (IDB)—built out of rules, and integrity constraints (IC). I will also assume that there is a cooperative layer between the database and a user where e-scenarios are stored and processed. Let us consider a simple (toy) example of such a database presented in Table 1.

As it might be noticed IDB contains rules for the database. Also new concepts might be introduced here (see the concept $locusr$ in Table 1). E-scenarios stored in the cooperative layer are built on the basis of IDB rules (IDB rules are used as premises).

$$
\begin{array}{c}
?locusr(a_i) \\
locusr(a_i) \to usr(a_i) \\
locusr(a_i) \to live(a_i, p) \\
usr(a_i) \wedge live(a_i, p) \to locusr(a_i) \\
?\{locusr(a_i), \neg locusr(a_i), usr(a_i)\} \\
?usr(a_i) \\
\begin{array}{cc}
\begin{array}{c}
usr(a_i) \\
?live(a_i, p) \\
\begin{array}{cc}
\begin{array}{c} live(a_i, p) \\ usr(a_i) \wedge live(a_i, p) \\ locusr(a_i) \end{array} &
\begin{array}{c} \neg live(a_i, p) \\ \neg locusr(a_i) \\ \end{array}
\end{array}
\end{array} &
\begin{array}{c}
\neg usr(a_i) \\
\neg locusr(a_i) \\

\end{array}
\end{array}
\end{array}
$$

Figure 1: Schema of an e-scenario for a question of the form 'Is a_i a local user?'

For example a relevant e-scenario for a question of the form 'Is a_i a local user?' would fall under the scheme presented in Figure 1.

The following logical facts were used in designing this e-scenario:

1. $\mathsf{Im}(?A, C \to A, ?\{A, \neg A, C\})$
2. $\mathsf{Im}(?\{A, \neg A, C\}, ?C)$
3. $\mathsf{Im}(?A, B_1 \wedge B_2 \to A, B_1, A \to B_2, ?B_2)$

As e-scenarios are stored in the cooperative layer between a user and the database, each question of a user might be processed and analysed against these e-scenarios. I will consider two types of user's questions: (i) about facts (i.e. concerning EDB part) and (ii) about concepts introduced in the IDB part of the database. In both cases the procedure would be the same. The task will be to find user's query among e-scenarios stored in the cooperative layer. When a query is found, its position in the e-scenario should be checked. There are two possibilities:

1. user's question is one of the queries of the e-scenario (in our example, e.g. questions of the form $?live(a_i, p)$);
2. user's question is the initial question of the e-scenario (e.g. questions of the form $?locusr(a_i)$).

Now—on the basis of this search—we may generate two types of question-response before executing user's query against the database:

1. *Were you aware that your question depends on the following questions ...? Would you also like to know their answers?*
   This question-response allows a user to decide how many details he/she wants to obtain in the answer. This also might be potentially useful for future search.
2. *Your question influences a higher level question. Will you elaborate on this subject (follow search in this topic)? May I offer a higher level search?*

The procedure might be described as follows:
E-scenarios $\{\Phi_1, ..., \Phi_n\}$ ($n \geqslant 1$) are stored in the cooperative layer. By the uQ we designate user's question. Q_i is the initial question of Φ_i (i.e. the root of Φ_i). Q_i^* is an e-node of Φ_i. The procedure of generating question-responses is the following.

- For each Q_i (where $1 \leqslant i \leqslant n$) check if $uQ = Q_i$.
  - If $uQ = Q_i$, then uQ is a question about a concept introduced in IDB. Return a question-response of the first kind.
  - To report all questions influencing Q_i, list all Q^* of Φ_i.[4]
- If $uQ \neq Q_i$, then for each Q_i^* (where $1 \leqslant i \leqslant n$) check if $uQ = Q_i^*$.
  - When you identify $Q_i^* = uQ$, then return a question-response of the second kind.
  - To report higher level question, return Q_i of Φ_i.

Now let us consider some simple examples of questions evaluated against the exemplary database (Table 1).

Example 1 uQ_1*: Is c a local user? (*$?locusr(c)$*).*

On the basis of a schema presented in Figure 1 we generate an e-scenario for question $?locusr(c)$ *by substituting c for* a_i*. Let us refer to this e-scenario as* Φ_1*. Consequently we will refer to its initial question* $?locusr(c)$ *as* Q_1*.*

In this case $uQ_1 = Q_1$ *so the procedure will generate a question-response of the first type. To report all questions on which* uQ_1 *depends the procedure returns all queries of* Φ_1*, i.e.:* $?usr(c)$ *and* $?live(c, p)$*. So the response in the natural language form would be:*
Were you aware that your question depends on the following questions: 'is c *a user?' and 'does* c *live in* p*?'? Would you also like to know their answers?*

Example 2 uQ_1*: Does c live in* p*? (*$?live(c, p)$*).*

Also in this case we will use Φ_1*. This time* $uQ \neq Q_1$*, so the procedure tries to match uQ with queries of* Φ_1*. Such matching is successful for* Q_1^**:* $?live(c, p)$*. The higher level question reported in question-response will be simply* Q_1*. So the response in the natural language form would be:*
Your question influences a higher level question: 'is c *a local user?'. Will you elaborate on this subject (follow search in this topic)? May I offer a higher level search?*

What is important, all the necessary data needed to generate question-responses of the analysed kinds might be obtained on the basis of e-scenarios analysis before their execution against the database (i.e. are done in the cooperative layer).[5]

Of course it might be the case that user's question will be identified in more than one e-scenario. Then presented question-responses should report all e-scenarios found. This will have the effect that a user will be aware of contexts in which his/her query is involved in the database. As these are question-responses, the system is expecting user's answer. This answer might be negative, i.e. a user will not use proposed suggestions. In this way the interaction with the system will be closer to the natural language interaction.

[4] In the case of really big e-scenarios (with many auxiliary questions) it would be useful to allow a user to decide how many questions influencing Q_i to report.

[5] It is worth to mention that also other techniques of cooperative answering might be used with e-scenarios (after their execution)—cf. [Łupkowski, 2010].

Summary

The proposed simple technique of generating question-replies will enrich cooperative interfaces with question posing capability. What is important, functionality offered by question-replies will be analogical to the functionality of cooperative answers, i.e. it will:

- inform a user (in an indirect manner) about the database schema (this will influence his/her future search and should allow to avoid wrongly formulated questions);
- adjust the level of generality of provided answers to the user's current needs;
- personalise the user's questioning process.

The advantage of question-replies usage will also be a decreased number of database transactions as the question analysis and operations on e-scenarios will be performed in the cooperative layer. As such, question-responses will also supplement the framework proposed in [Łupkowski, 2010].

Last but not least, involving questions into the cooperative answering process is motivated with natural language dialogues. As a result, interactions with databases and information systems may become more 'natural' and somehow closer to the real-life conversations. Already mentioned corpus study [Łupkowski and Ginzburg, 2013] reveals that in natural language dialogues question-responses addressing the issue of the *way* the answer to the initial question should be given are commonly met. This kind of a question given as an answer might be observed in the following example:

> A: Okay then, Hannah, what, what happened in your group?
> B: Right, do you want me to go through every point? [K75, 220–221]

As it might be noticed, question-replies proposed in this paper will fit this schema.

References

Benamara, F. and Saint-Dizier, P. [2003], WEBCOOP: a cooperative question-answering system on the web, *in* 'Proceedings of the tenth conference of European chapter of the Association for Computational Linguistics'.

Gaasterland, T., Godfrey, P. and Minker, J. [1992], 'An overview of cooperative answering', *Journal of Intelligent Information Systems* **1**, 123–157.

Gal, A. [1988], Cooperative Responses in Deductive Databases, PhD thesis, University of Maryland, Department of Computer Science.

Ginzburg, J. [2010], Relevance for dialogue, *in* P. Łupkowski and M. Purver, eds, 'Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue', Polish Society for Cognitive Science, Poznań, pp. 121–129.

Godfrey, P. [1997], 'Minimization in cooperative response to failing database queries', *International Journal of Cooperative Information Systems* **6**(2), 95–149.

Łupkowski, P. [2010], Cooperative answering and inferential erotetic logic, *in* P. Łupkowski and M. Purver, eds, ‘Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue’, Polish Society for Cognitive Science, Poznań, pp. 75–82.

Łupkowski, P. [2012], Erotetic Inferences in Natural Language Dialogues, *in* ‘Proceedings of the Logic & Cognition Conference’, Poznań, pp. 39–48.

Łupkowski, P. and Ginzburg, J. [2013], A corpus-based taxonomy of question responses, *in* ‘Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)’, Association for Computational Linguistics, Potsdam, Germany, pp. 354–361.
URL: *http://www.aclweb.org/anthology/W13-0209*

Motro, A. [1994], ‘Intensional answers to database queries’, *IEEE Transactions on Knowledge and Data Engineering* **6**(3), 444–454.

Urbański, M. and Łupkowski, P. [2010], Erotetic search scenarios: Revealing interrogator’s hidden agenda, *in* P. Łupkowski and M. Purver, eds, ‘Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue’, Polish Society for Cognitive Science, Poznań, pp. 67–74.

Webber, B. L. [1985], Questions, answers and responses: Interacting with knowledge-base systems, *in* M. Brodie and J. Mylopoulos, eds, ‘On Knowledge Base Management Systems’, Springer, pp. 365–401.

Wiśniewski, A. [1995], *The Posing of Questions: Logical Foundations of Erotetic Inferences*, Kluwer AP, Dordrecht, Boston, London.

Wiśniewski, A. [2001], ‘Questions and inferences’, *Logique et Analyse* **173–175**, 5–43.

Wiśniewski, A. [2003], ‘Erotetic search scenarios’, *Synthese* **134**, 389–427.

Wiśniewski, A. [2010], ‘Erotetic logics. internal question processing and problem-solving’. URL=`http://www.staff.amu.edu.pl/~p_lup/aw_pliki/Unilog`.

Paweł Łupkowski
Department of Logic and Cognitive Science,
Adam Mickiewicz University
Poznań, Poland
e-mail: `Pawel.Lupkowski@amu.edu.pl`
URL: `http://amu.edu.pl/~p_lup/`