A Brief Overview of Cooperative Answering

F. Corella K. P. Lewison

Revised August 20, 2009

Abstract

A cooperative response to a query is an indirect response that is more helpful to the user than a direct, literal response would be. Interest in cooperative responses arose in the context of natural-language query answering, but the concept carries over, with suitable modifications, to a formal-language setting. Cooperative answering has been studied over the last thirty years, but, until recently, work has mostly been confined to academia. This is changing now, however, and we argue that cooperative answering has an important role to play in the surge of innovation that is now taking place in search and information retrieval.

Preface

Noflail Search, available at noflail.com, is a search engine that provides cooperative responses to queries that produce zero results. Such queries are rare on the Web at large, but more frequent when search is restricted to a particular Web site, something that is easy to do in Noflail Search. This Pomcor white paper provides a brief overview of cooperative answering as background information for the cooperative answering feature of Noflail Search.

1 Introduction

In an information retrieval system, a direct, literal response is not always the best answer to a query. This is particularly the case when the set of results produced by the query is empty. A *cooperative response* is an indirect response that is more helpful to the user than the direct response would be.

Cooperative responses emerged in the context of natural-language question-answering. They were originally motivated by the desire to follow the conventions of human conversation in human-machine interactions carried out in natural language. When a formal language, is used rather than natural language, or no language is used, this original motivation is not present but certain cooperative responses are nevertheless appropriate and useful [18, 17, 14, 3, 20]. A cooperative response may explain the failure of a query to produce results and/or suggest follow-up queries. In the case where a query does produce results, a cooperative response may provide additional information not explicitly requested by the user.

In this white paper we recall the origins of cooperative answering thirty years ago and mention some of the academic research that followed, giving a sample of pointers to the literature.[1] Most of the work on cooperative responses has been so far confined to academia, but this is now changing. There is now a surge of innovation related to search and information retrieval, in which cooperative responses should play an important role. We point out recent developments that show how this is beginning to happen.

2 The Origins of the Concept

Kaplan [16] is often credited with being the first to implement a system of cooperative responses to natural-language queries; at about the same time Janas [14] provided cooperative responses to database queries formulated in the relational calculus. Kaplan argued that a natural language interface should follow some of the conventions of human dialog, and showed how an interface that did not follow those conventions would appear to stonewall or otherwise seem unnatural. Here are some of the striking examples he used to make the point (with numbering and captions added by us for reference within this paper):

```
Example 1
  a. Dialog with a non-cooperative interface
    Q: Which students got a grade of F in CIS500 in Spring '77?
    R: Nil.      [the empty set]
    Q: Did anyone fail CIS500 in Spring '77?
    R: No.
    Q: How many people passed CIS500 in Spring '77?
    R: Zero.
    Q: Was CIS500 given in Spring '77?
    R: No.
  b. Dialog with a cooperative interface
    Q: Which students got a grade of F in CIS500 in Spring '77?
    R: CIS500 was not given in Spring '77?

Example 2
  a. Dialog with a non-cooperative interface
    Q: Are there any seats available in the orchestra for tonight's
       Rolling Stones' concert?
    R: No.
    Q: Are there any in the balcony?
    R: Yes.
  b. Dialog with a cooperative interface
    Q: Are there any seats available in the orchestra for tonight's
       Rolling Stones' concert?
```

[1] This white paper does not pretend to be a survey. Our intention is only to give an idea of the work that has been done. A survey of early work can be found in [8].

```
    R: No, but there are some in the balcony.

Example 3
  a. Dialog with a non-cooperative interface
    Q: What are the phone numbers of managers in marketing?
    R: 293-4958, 584-7945, 293-7754...
  (Kaplan let the reader imagine a more cooperative answer.)
```

Cooperative response 1b corrects a false presupposition of the user, and cooperative response 2b anticipates a follow-up query. A cooperative response to question 3a would provide information not explicitly requested by the user, viz. the names of the managers paired with their phone numbers.

3 Research on Cooperative Responses

Kaplan [16], working in a natural-language setting, emphasized the need to correct wrong presuppositions held by the user. Janas [14] worked in a formal-language but was also was concerned with correcting wrong assumptions made by the user. But it was observed by Corella et al. [3] and Motro [20] that cooperative responses are appropriate even in cases where it is not possible to attribute any assumptions to the user. In [3], for example, cooperative responses were produced in the context of a bibliographic database. No natural language was used. Queries were Boolean combinations of search terms, each search term specifying a condition on a particular database field. It did not seem possible to infer any user-held assumptions from such queries. And yet, when a query produced no database records, it was clearly useful to suggest more general queries that did not produce results, as well as more general queries that did produce results. The former provided an explanation of the failure, even though they could not be said to correct any false presuppositions in the user's query. Generalizations of the query were subqueries obtained by removing search terms; the cooperative response listed the maximal generalizations (subqueries with minimal sets of terms) that did not produce results, as explanations of the failure, as well as the minimal generalizations (subqueries with maximal sets of terms) that did produced results, as follow-up queries.

Kaplan used no domain knowledge to provide cooperative responses, besides the database itself and a lexicon. Other researchers however have used domain knowledge, expressed in a variety of formalisms.

Domain knowledge may be provided for the express purpose of generating cooperative responses, or it may already be available for other purposes. As an example of the latter, a database may have integrity constraints, which can be viewed as a form of domain knowledge. The fact that a query conflicts with an integrity constraint may be an indication of an incorrect presupposition held by the user about the contents of the database. A cooperative response can then be produced to correct the false presupposition. As a second example, deductive databases allow for the use of taxonomies in answering queries. The same taxonomies may be used to generalize a query that produces no results.

Janas [14] used integrity constraints in the context of a relational database, but only for the purpose of simplifying queries. Gal and Minker [12] used integrity constraints, formulated within a deductive database formalism, to compute appropriate cooperative responses. Gaasterland et al [9] used taxonomies, again in the setting of a deductive database. Motro [21] used integrity constraints and completeness assertions encoded as relational database tuples using an ad-hoc convention. Kao [15] used a rich variety of meta-data encoded in RM/T (Codd's relational-model extension), including entity relationships, entity-relationship constraints, and several kinds of hierarchies. Chu et al. [2] used Type-Abstraction Hierarchies (TAH) obtained by clustering the values of both numeric and non-numeric attributes.

One difficulty in producing cooperative responses is that the number of possible generalizations of a given query is very large. Take the simplest case where the query is a conjunction of search terms, and generalizations are subqueries obtained by removing terms: the number of possible generalizations of a query with N terms is $2^N - 2$.

This means that, given a query that fails to produce results, a cooperative response that lists the maximal generalizations that also fail, or the minimal generalizations that succeed, may be very large. And even if it is not large, it may be hard to compute. From a complexity-theoretical point of view, Godfrey [13] showed that finding all maximal failing generalizations, or all minimal succeeding generalizations, is an intractable problem, whereas finding a fixed number of them is polynomial in the number of terms N of the query. He proposed an algorithm that enumerates such generalizations, finding the first ones quickly before slowing down. This algorithm could be used to produce a partial cooperative response that grows with time, with new subqueries being added to the response while the user examines the first few ones.

When domain knowledge is used to compute cooperative responses, even a query with a single search term may have many possible generalizations. It is then necessary to determine what information should be included in the response. Several false presuppositions may be detected, but some of them may be more critical than others. Many generalizations or other query relaxations may be found, but some of them may be more relevant than others. Gal [11] proposed a collection of heuristic rules for selecting pertinent information while constructing a cooperative response. Gaasterland [7] used constraints explicitly specified by the user to rule out inappropriate query relaxations, and heuristics to order the remaining ones. More recently, Muslea [23] has used a machine learning algorithm to produce cooperative responses. Remarkably, the algorithm is used at run-time, every time a query fails to produce results. It inspects a randomly chosen subset of the database and learns relationships between existing data that it uses to relax the query.

Most research in the field of cooperative answering has focused on using as a cooperative response a list of alternative queries (with or without their answers), but other types of cooperative responses have been proposed as well. Kaplan's CO-OP provided additional information not requested by the user, something essential in a natural language interface since queries in natural language often do not fully specify the information being requested. Gaasterland [7] and Minker and Gal [19] also provided additional information. Motro [22] provided intensional responses. Gaasterland and Lobo [10] provided qualified responses. Benamara and Saint-Dizier provided cooperative responses to Web search queries that pointed to relevant know-how through hyperlinks [1]. More recently, query relaxation has been studied in the context of searching XML data [6].

4 Thirty Years Later

Thirty years have passed since Kaplan's thesis, and an impressive amount of research work has been done on cooperative answering. Most work so far has been confined to academia[2], but this is now changing.

We are witnessing today a multi-faceted transformation of the field of search and information retrieval. Aspects of this transformation include: the birth of a search ecosystem where independently developed search components can be integrated through Web APIs; fast-paced innovation in the area of user interfaces; and the development of multiple ways of using semantics to improve search. Every day brings the announcement of several new search engines, as can be seen, for example, by subscribing to AltSearchEngines. A new ACM annual conference on Web Search and Data Mining was started in 2008 to serve as a forum for discussion of all the new ideas being generated.

Cooperative answering should play an important role in this surge of innovation, as is beginning to do so.

At least one important database system, the Clinical Trials datbase of the National Library of Medicine, available at ClinicalTrials.gov, provides a form of cooperative responses. Specifically, when a conjunction of search terms produces zero results the user interface suggests all the succeeding subqueries of the user's query as follow-up queries.

Noflail Search [5, 4], a search front-end available at noflail.com, provides cooperative responses to Web searches that fail to produce results. Such queries are rare on the Web at large, but more frequent when search is restricted to a particular Web site, something that is easy to do in Noflail Search.

Noflail Search is a piece of the new search ecosystem, as it obtains search results from the Bing™ API[3]. To compute cooperative responses through a Web

[2] In 1997, Godfrey [13] observed that no commercial databases provided cooperative responses. He attributed this fact to the intractability of the problem, but we believe it should rather be seen as failure of technology transfer with no compelling technical reasons. Performance was not a problem even in 1981, when the cooperative answering described in [3] was implemented on a real-world bibliographic database under development. The code was not put into production because the project was completed and everyone involved had moved on before the database system itself was completed. The first author of this paper must take responsibility for not collecting performance results, while the project was in progress, for inclusion in the subsequent publication. Unfortunately, other researchers have not published performance results either. Performance results for cooperative responses in Noflail Search are available in a Pomcor technical report [4].

[3] Bing is a trademark of Microsoft Corporation. Pomcor is not affiliated with Microsoft.

API we had to solve a new technical problem. Typical API latencies range from one tenth of a second to one or two seconds. Such latencies are acceptable for running a single query, but make it impractical to compute a cooperative response, which requires running many subqueries, using the sequential algorithms that have been proposed so far in the literature. We solved the problem by devising a parallel algorithm that takes advantage of the inherent parallelism offered by a Web API [4].

There are many opportunities today for further work on cooperative responses, ranging from user interface design to semantics. The use of semantics to construct cooperative responses is a particularly promising prospect. Researchers used semantic domain knowledge early on, but they often had to create the semantic data that they used. Today real-world semantic data is available in various forms, including ontologies such as the Unified Medical Language System (UMLS) of the National Library of Medicine, semantic markup of Web pages using microformats or RDF, or even natural-language repositories of information such as Wikipedia. Hakia has announced a commercial ontology which presumably will be available to third parties in the search ecosystem. Real-world semantic information should allow a more in-depth investigation of semantics-based cooperative answering. We believe that cooperative answering will make a major contribution to the on-going search revolution.

References

[1] F. Benamara and P. Saint-Dizier. Advanced relaxation for cooperative question answering. In M. T. Maybury, editor, *New Directions in Question Answering*, pages 263–274. AAAI Press, 2004.

[2] W. W. Chu, H. Yang, K. Chiang, M. Minock, G. Chow, and C. Larson. Cobase: A scalable and extensible cooperative information system. *J. Intell. Inf. Syst.*, 6(2/3):223–259, 1996.

[3] F. Corella, S. J. Kaplan, G. Wiederhold, and L. Yesil. Cooperative responses to boolean queries. In *ICDE*, pages 77–85. IEEE Computer Society, 1984.

[4] F. Corella and K. P. Lewison. A parallel algorithm for computing cooperative responses through a web api. Technical report, Pomcor, August 2009. Available at the Pomcor Web site.

[5] F. Corella and K. P. Lewison. Searching the web more effectively with multiple simultaneous queries. Presented at the Search Engine Meeting, April 2009. White paper available at the Pomcor Web site.

[6] G. Dong, X. Lin, W. W. 0011, Y. Yang, and J. X. Yu, editors. *Advances in Data and Web Management, Joint 9th Asia-Pacific Web Conference, APWeb 2007, and 8th International Conference, on Web-Age Information*

Management, WAIM 2007, Huang Shan, China, June 16-18, 2007, Proceedings, volume 4505 of *Lecture Notes in Computer Science*. Springer, 2007.

[7] T. Gaasterland. Cooperative answering through controlled query relaxation. *IEEE Expert*, 12(5):48–59, 1997.

[8] T. Gaasterland, P. Godfrey, and J. Minker. An overview of cooperative answering. *J. Intell. Inf. Syst.*, 1(2):123–157, 1992.

[9] T. Gaasterland, P. Godfrey, and J. Minker. Relaxation as a platform for cooperative answering. *J. Intell. Inf. Syst.*, 1(3/4):293–321, 1992.

[10] T. Gaasterland and J. Lobo. Qualifying answers according to user needs and preferences. *Fundam. Inform.*, 32(2):121–137, 1997.

[11] A. Gal. *Cooperative responses in Deductive Databases.* PhD thesis, University of Maryland, College Park, 1988.

[12] A. Gal and J. Minker. Informative and cooperative answers in databases using integrity constraints. In *Natural Language Understanding and Logic Programming Workshop*, pages 277–300, 1987.

[13] P. Godfrey. Minimization in cooperative response to failing database queries. *Int. J. Cooperative Inf. Syst.*, 6(2):95–149, 1997.

[14] J. M. Janas. On the feasibility of informative answers. In *Advances in Data Base Theory*, pages 397–414, 1979.

[15] M. Kao, N. Cercone, and W.-S. Luk. Providing quality responses with natural language interfaces: The null value problem. *IEEE Trans. Software Eng.*, 14(7):959–984, 1988.

[16] S. J. Kaplan. *Cooperative responses from a portable natural language data base query system.* PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 1979.

[17] S. J. Kaplan. Designing a portable natural language database query system. *ACM Trans. Database Syst.*, 9(1):1–19, 1984.

[18] R. M. Lee. Conversational aspects of database interactions. In S. B. Yao, editor, *VLDB*, pages 392–399. IEEE Computer Society, 1978.

[19] J. Minker and A. Gal. Producing cooperative answers in deductive databases. In P. Saint-Dizier and S. Szpakowics, editors, *Logic and Logic Grammar for Language Processing.* Ellis Horwood, New York, 1990.

[20] A. Motro. Query generalization: A method for interpreting null answers. In *Expert Database Workshop*, pages 597–616, 1984.

[21] A. Motro. Seave: A mechanism for verifying user presuppositions in query systems. *ACM Trans. Inf. Syst.*, 4(4):312–330, 1986.

[22] A. Motro. Intensional answers to database queries. *IEEE Trans. Knowl. Data Eng.*, 6(3):444–454, 1994.

[23] I. Muslea and T. J. Lee. Online query relaxation via bayesian causal structures discovery. In M. M. Veloso and S. Kambhampati, editors, *AAAI*, pages 831–836. AAAI Press / The MIT Press, 2005.