NL Interfaces and the Turing Test*

John Nerbonne
nerbonne@let.rug.nl

Alfa Informatica and
Centre for Behavioral, Cognitive and Neuro-sciences
Rijksuniversiteit Groningen
Oude Kijk in 't Jatstraat 26
P.O.Box 716
9700-AS Groningen
The Netherlands

Abstract

A successful natural language interface (NLI) would carry on a conversation with a user in much the same way that a human being would—in other words, it would constitute a proof of machine intelligence of the sort required in the Turing test. NLIs have been excellent research vehicles but impractical in application. Too many applications and application plans assume an approximation of the Turing test level of performance, even though it remains quite remote. NLIs with more modest goals, and especially those which aim to supplement, rather than replace alternative interfaces, may nonetheless be useful.

1 The Turing Test and Natural Language

In 1950 Alan Turing (Turing 1963) proposed that we might regard a machine as intelligent if it could conduct a substantial conversation with a human interlocutor as well as a human being could—well enough so that the interlocutor could not reliably tell the difference. This is the famous Turing test, and although it has its critics (Searle 1984), it is widely regarded as fair—the problem of simulating all the abilities needed to conduct a conversation is sufficiently varied and challenging that one does indeed regard it as a proof of intelligence.

Computational linguistics (CL) has in general benefitted—both in its content and in its funding—from the view that communication is essential to intelligence. Language research has occupied a central position in artificial intelligence (AI) and has received from AI both important research techniques (e.g., inheritance-based reasoning—see Daelemans et al. 1992) and significant research funding. This is entirely appropriate given the scientific goal of AI—to characterize and probe the limits of computational intelligence. In this context language research has often functioned as a vehicle with which to test theories and techniques—a bit like a Carnot machine in thermodynamics—and it has served its purpose well.

The research and development of natural language interfaces (NLIs) has received enormous impetus from the correct perception that the implementation of an NLI requires varied and substantial human-like intelligence. An NLI as usually conceived should be able to carry on an intelligent conversation with a user, much as a human with similar informational resources might. But this is just to say that NLIs are rough attempts at building software capable of passing Turing's test of intelligence—that of maintaining an extended natural conversation with a human being.[1]

The research community is agreed on the complexity of the problem. But it is still worth reviewing the sorts of information re-

*The ideas for this paper developed in discussions with Stephan Busemann, Elizabeth Hinkelman, Klaus Netter, Günter Neumann, Stephan Oepen, Stephen Spackman, Harald Trost, and Hans Uszkoreit, none of whom necessarily agrees with them. I am grateful to Bert Bos and Ion Androutsopoulos for comments on an earlier draft.

[1] While some of Turing's examples make it clear that he intended the conversations to be wide-ranging (he makes cultural allusions and attempts fallacious arguments which the machine must debunk in one of the example dialogues), I do not believe that any of the further abilities need be regarded as essential.

sources that must be brought to bear in interpreting utterances in natural language, if only to reinforce our appreciation of the enormity of the task.

linguistic knowledge of several sorts is needed to characterize the input and its meaning. There are several subareas, and research is not closed in any of them.

- morphological and lexical
- syntactic
- semantic
- pragmatic

immediate discourse context including perhaps extralinguistic context is needed to resolve anaphoric reference and interpret fragmentary expressions.

domain knowledge is needed to link speaker meanings with application semantics, but also often for disambiguating speaker meanings. Note that this resource must include inferential capabilities in order to be useful.

Information resources are listed above whenever there is substantial agreement that they would be need in order to conduct effective, albeit primitive conversation.[2] we shall refer to this list of requirements as the CORE REQUIREMENTS, since it will turn out that the justification of NLIs as practical endeavors hinges on meeting these requirements.

The list has not been exaggerated to emphasize how difficult things are. On the contrary, one often finds much more demanding catalogues of desirable properties for NLIs. I shall ignore all of these below, since it is likely that usable NLIs could be developed without them. But the list is interesting, both as a checklist of how we are doing vis-à-vis the Turing test, and as a reminder that things are far from perfect even after core requirements are met.[3]

intentional models or **task models** are relevant in determining the appropriate domain action to consider in case this is not specified in the literal meaning of what is said. Inference is essential here as well.

commonsense models or world knowledge may disambiguate. Inference is again needed. Cf. Schank and Riesbeck 1981.

discourse segmentation facilities model task structure more seriously and may influence interpretation (cf. Grosz and Sidner 1986) as well as allow smooth mixing of initiative (Walker and Whittaker 1990), e.g., in order to enable the initiation of a clarifying subdialogue where needed.

user modeling including users' experience, degrees of expertise, etc. may be needed to forestall errors and accommodate users better. Cf. Kobsa and Wahlster 1988.

flexibility in dealing with unknown words and structures increases robustness.

knowledge of likely errors helps characterize recover strategies.

Each of these problems defines more than enough material for entire research fields, and it is worth remembering that there need not exist usable solutions at all. That is, even once we have understood some of these areas better, there is no guarantee that the inferences and computations involved will be tractable or even decidable. We just do not know. And the fact that there are so many unsolved problems and problem areas makes the enterprise especially sensitive. Failure could arise from any number of different sources.

Finally, if each of these topics were not in itself daunting, there would remain the coordination problem. As Allen et al. 1989 note: "no one knows how to fit all of the pieces together."

Let me clarify the argument at the risk of repetition: the argument does NOT seek to establish that non-core requirements must be met in order to have useful NLIs. Rather, we shall claim that core requirements are impeding practical NLIs to an extent that keeps them out of the market. We list the others as a reminder of all the other factors

[2] I formulate the condition this way, aware that CL is a lively and even contentious field, and that each of the areas might be challenged. Some semantic grammars (Schank and Riesbeck 1981) deny the usefulness of syntax and morphology, for example. Still there is large consensus today that this much is needed.

[3] For the non-NL specialist it may be worth mentioning that progress in solving the non-core problems lags considerably. The problems are novel and hard.

that arise, at least potentially, when an NLI is to be used.

While this research has led to important insights about language and intelligence vis-à-vis computation, it is not surprising that it has failed in achieving its nominal goal. It would be much more surprising to see it succeed in the near term.

2 Why NLI Products?

As I emphasized above, there is no problem with identifying a research vehicle which is too difficult to realize using present technology. But somewhere along the line language researchers seem to have begun believing that the research vehicle was a product prototype, and that useful machine conversationalists were a realistic possibility.

Most of this effort was spent into attempting to build NLIs to database (DB) query systems, where my own experience also lies. So this is the area that I have most firmly in mind in making the remarks below. I believe the considerations remain valid, *mutatis mutandis* for NLIs to other software systems (help systems, appointment managers, spreadsheets, etc.), but let us note that an element of analogy infects the argument. NLIs for DB query were from the start an attractive target for the natural language (NL) technology under development.

It is easy to understand why. A fully successful NL would provide an (i) expressive and yet (ii) concise interface which requires (iii) no training in a particular programming language and (iv) no familiarity with particular data structures and program organization. Using NL, one can say all one wants, and one can say it briefly, without learning a specialized code, and without studying how the program has been organized. The technology promises to make computational resources much more accessible than they normally are.

In order to fulfill this promise, the core requirements noted above must be met. The linguistic knowledge must be present, or the system will be limited to a subset of NL, and it will turn out that training is required after all. The system must be sensitive to discourse context first because users will speak in terms dependent on context (and will require training if they are to avoid it). This point has been been confirmed emphatically in user studies (Whittaker and Stenton 1989). A second reason for needing sensitivity to discourse is in order to realize the conciseness which NL offers. This conciseness turns on the ability of users to communicate using not only pronouns and other anaphoric devices, but also fragments. Finally, domain knowledge—meaning a reliable link between NL and the domain—is an absolute necessity if the NLI is to "hide" the implementation details of the application from the user.

It is crucial to note here that very impressive abilities are needed not just to make NLIs human-like—so that they can pass their Turing tests—but in order for them to be successful at all.

In one view, NLIs for DB query cleverly finesse one of the crucial general problems for NLIs, that of domain reasoning. NLIs to DBs finesse this problem because DBs are very intelligent applications. In particular, relational databases, the subject of the most intense experimental activity in this field, come with a guarantee of completeness—information may be queried successfully in any of a large number of ways because of the relational algebra the DB is based on. In principle, this relieves the NLI developer from a great deal of inference which the NLI itself would otherwise have to provide. Interfaces to other software systems should therefore be expected to be more difficult, *ceteris paribus*.[4] Thus, considered as examples of the general problems of NLIs, DB query interfaces represent a very welcome simplification.

Of course, the core requirements need only be met with respect to whatever application is under consideration. An interface to a personnel database need not understand the language of sales or repairs. But general and easily implementable (optimally, automatable) methods are needed if NLI technology is to be economically viable. These are substantial tasks.

Estimations of the development of the technology were once nonetheless quite op-

[4] In particular, this will be true for systems of the size and complexity of databases. We return below to the difficulties of motivating NLIs for simple applications.

timistic. We find assessments such as the following:

> By 1987 a natural language interface should be a standard option for users of DBMS and 'Information Centre' type software, and there will be a reasonable choice of alternatives.
> (Johnson 1985, p.14)

2.1 Problems

But none of the potential advantages of NL technology is realized in the current state of the technology:

expressive capacity of NLIs is hampered by missing and inaccurate coverage in linguistic knowledge and elsewhere.

We do not even have effective gauges for (i) how well systems analyze a given NL, e.g. English; (ii) what coverage is required for a given application; or even (iii) how the coverage of different systems compares.

There are of course efforts in this direction for some of the specializations, especially syntax. (See Nerbonne et al. 1993 and references there for further information.) But for most of the technology there is no way of evaluating quality or of assessing usefulness.

conciseness of NLIs is limited by missing discourse resolution capabilities. What makes NL concise is exactly its ability to use context to support anaphoric and fragmentary expression. But this area continues to resist substantial progress. Virtually the only area with any degree of measurable success is that of simple pronoun resolution, where results are less than encouraging;[5] areas such as VP ellipsis, gapping, $\bar{\mathbf{N}}$ anaphora, coordination and comparative ellipses have not even matured to the point where good algorithms are available. The interpretation of fragmentary utterances remains fairly *ad hoc.*

naturalness —the feature that NLIs do not require user training—is not confirmed in practice. This is the inevitable consequence of an imperfect system. Androutsopoulos 1993 finds this the "most frequent complaint" against NLIs, citing studies by Tenant et al. 1983, Hendrix 1982, and Cohen 1991.

This problem often appears under the name of HABITABILITY (Watt 1968)—how easily users become accustomed to the inevitable limitations of the NLI. If users could adjust easily to a flawed system, they could perform useful work with it. The reports cited above indicate that this is not happening.

transparency refers to the claim that NLIs free users from needing to know the specific organization of data (in addition to being freed from needing to know how to formulate queries and commands).

This advantage is seldom, if ever, realized even in toy systems because there is no standard methodology for connecting to databases, in spite of dozens of experimental systems. See Iida et al. 1989 for a discussion of the problems in trying to guarantee that a NL lexicon is complete for a large database.

See Moore 1982 (and other papers in that ACL panel) for a discussion of the semantic problems associated with guaranteeing transparency. Although there is consensus, e.g., that the correct solution is outlined in the contribution by Warren 1982 (in that same ACL panel as Moore 1982)—outlining a deductive component needed to bridge mismatches in user and DB conceptual schemas—this has not matured into a standard methodology. In particular, it is impossible to determine when the knowledge base of the deductive component is complete.

The failure of NLIs to solve the problems of application interface have meant that virtually all commercial systems aim at a very limited class of databases—those with personnel or sales information.

Summarizing, in spite of substantial effort and genuine progress, a general and com-

[5] Two state of the art resolution algorithms (with some variations) for simple pronouns are compared in Walker 1989. The results vary greatly depending on type of text and are significantly worse for task dialogues of the sort NLIs are designed for, where they vary from 50-65% accuracy.

mercially viable NLI technology remains distant.

2.2 Competition

Androutsopoulos 1993 reminds us of the important shift in background which has occurred since NLIs were first considered as general software interfaces. As NLIs sought practical success, some of the greatest problems lay not in their own technology, but in the phenomenal success of their competitors—graphical and form-based user interfaces (GUI).

Before the advent of GUIs, the usefulness of NLIs was often seen to lie in PARTICULAR USER TYPES. Users who were not computer specialists and who were infrequent users were seen as the key to a commercially viable NLI technology (Petrick 1976, Shneiderman 1980). But these users are in general (perhaps not always) served better by GUIs, which are more reliable and less costly in installation—often automatically available in commercial DBMS systems.

The advent of GUIs and later interface toolkits (Apple Computer 1988) changes the economics of this argument completely, raising the specter of comparing NLIs representing millions of research dollars against simple GUIs built in a few person-months. The former are complicated to install, not easily transported (in spite of improvements—see Martin et al. 1983 and Androutsopoulos et al. 1993), and have significant habitability problems, as indicated above). The latter are often installed automatically, transported easily, and popular with users.

Even after GUIs had been identified as the important competition, NL researchers saw important advantages for NLIs over GUIs. Perrault and Grosz 1988 (p.134) are typical of the first reaction of the NLI research community to GUIs—sobered, but optimistic. Although I shall argue that their characterization needs reassessment, it is instructive. They suggest that NLIs will be more appropriate than GUIs for PARTICULAR APPLICATION TYPES, namely those meeting the following criteria:

complex information In this case the information is not easily presented in forms or in immediately intuitive graphics.

nonintuitive encoding of information. In these applications the NLI allows the user to operate at the "knowledge level". For example, a DB will normally encode supervisor relations in a single way—perhaps by listing a supervisor for each employee, or by listing a department for each employee and a supervisor for the department. The nonredundant encoding helps to ensure consistency. But users will query information unaware of the exact form in which it appears.

In these cases the NLI functions as a translator from the "commonsense" formulation of the problem to the application encoding.

very complex problem-solving It is difficult to specify the problem even in programming language interfaces.

The second point here commits a fallacy of the accident—if NLIs can answer queries using a more intuitive characterization of information than that used in the application, well then so can GUIs. Nothing hinges on the use of NL here. This is a more general point—that user interfaces ought to be prepared to operate at Newell's "knowledge level" (Androutsopoulos 1993, p.9), and that they must not rely on the application's encoding of data. In fact, it has been a normal design goal for all user interfaces for some time (Norman and Draper 1986, Peddie 1992, p.36).

Perrault and Grosz's other two points essentially up the ante: if NLIs are not better in general, then they are still better for the really hard cases—very complicated applications. It is not clear that this is correct, since problems related to the resolution of ambiguous, vague and underspecified information become exponentially worse as complexity increases. It is common to find elaborate systems providing hundreds of analyses for longer sentences. But it is certainly true that GUIs are cumbersome in complex applications, so let us concede for the sake of argument that Perrault and Grosz are right in the limit—that, when both technologies are mature and reliable, NLIs are superior to GUIs for dealing with complex information. Still, given the present state of NLIs the advantages they may eventually show in handling complex information cannot be practically realized. This is due to all the difficulties noted in Section 2.1. Realizing the potential advantage of NLIs in complex applications will require substantially better performance from NLIs.

2.3 Successful NLIs

In spite of all the technical problems we do find systems which appear to sell in modest numbers. Since I do not wish to endorse or discredit these, I will not mention any by name, but the systems I have in mind appear to be very selective in choosing which applications to try to interface to, and generally seem to regard personnel and sales databases as their "normal targets". I believe this is a reflection of the unsolved general problems in application interfaces referred to above (Section 2.1, "transparency").

In addition to commercial systems some of the ATIS demonstrators have shown very impressive performance (DARPA 1989 92), but these are for a very simple application (airline flight information).

3 Whither Practical NLIs?

NLIs remain an excellent research vehicle for artificial intelligence and for problems in computational linguistics, computational psychology, and experimental user interface construction. They are excellent research vehicles because of the many outstanding research problems which can be identified with one or another aspect of their operation. This is of course the same property which disqualifies them from great practicality in the foreseeable future.

A great deal of the effort spent on NLIs arguably has not had and should not have practical goals. I do not wish to dispute this.

But a question remains: is there any sense in attempting NLIs with practical goals in mind? While NLI efforts with practical goals must assume poor technical performance for some years to come, this may not stand in the way of all applications. And while GUIs will show cost/benefit superiority for probably even longer, this does not mean that NLIs have no useful function to serve.

3.1 If you can't lick 'em, ...

A key to identifying potentially practical NLIs is to concede the general battle to GUIs. GUIs are established successes and are unlikely to be overtaken by NLIs in any of the applications where they might compete—at least not in the near future. **In the near term NLIs can only be useful either in cooperation with GUIs—in multi-modal interfaces, or in ecological niches where GUIs are simply inapplicable.**

Several projects have combined NL and graphics or menu-based techniques for user interfaces (Bobrow et al. 1990, Cohen 1991, Neumann et al. 1993). JANUS (Bobrow et al. 1990) foresaw the use of menus and graphics in an NLI for disambiguation purposes. SHOPTALK (Cohen 1991) proceeds from a standard menu-based interface, but allows a user to fill menu slots not only with the standard menu items, but also using NL freely. COSMA (Neumann et al. 1993) is designed to allow the same use of NL in menu-slots we find in SHOPTALK.

This sort of application can also be clever in allowing NL without relying on it. For example, COSMA has a menu interface for graphics terminals, but allows users on dumb terminals to respond to appointment requests using NL. If the COSMA system can process the NL response, the appointment negotiation may proceed automatically. In case the NL processing fails, however, COSMA can queue the NL response to its human master (along with enough state information to allow him to get the appointment negotiation going again). This fallback allows a comfortable development path for NL to be exercised even as it improves.

And it will continue to be useful to consider application areas where GUIs are poorly suited. These may involve cooperation with speech processing (DARPA 1989 92) for applications in which either hands or eyes are busy, or applications which are remote from graphics-suitable terminals, but which could be supported by a modem connection, or applications in telephony or in the support of the handicapped. The COSMA appointment manager foresees application by users who travel and need to check their appointment calendars.

4 Summary

An ideal NLI would carry on a conversation with a user in much the same way that a human being would—in other words, it would constitute a proof of machine intelligence of the sort required in the Turing test. Of course this would be useful, but it is also difficult to produce. NLIs have been excellent research vehicles but impractical in application. Too many applications and application plans assume an approximation of the Turing test level of performance, even though it remains quite remote.

Unfortunately, it is not only the case that ideal NLIs need highly developed capabilities, but also that even modest applications need a great deal of presently unavailable ability.

Overwhelming competition from alternative and superior interface technologies has complicated the problems of NLIs. NLIs with more modest goals, and especially those which aim to supplement, rather than replace alternative interfaces, may nonetheless be useful. And the search for application areas in which alternative interfaces are technically infeasible, but in which NLIs could function, holds a further key to potential success for this technology.

References

Allen, J., S. Guez, L. Hoebel, E. Hinkelman, K. Jackson, A. Kyburg, and D. Traum. 1989. The Discourse System Project. Technical report, Computer Science, University of Rochester, November.

Androutsopoulos, I. 1993. Natural Language Interfaces to Databases: A Survey of the Current Technology. Unpublished document, Department of Artificial Intelligence, University of Edinburgh.

Androutsopoulos, I., G. Ritchie, and P. Thanisch. 1993. Masque/sql—An Efficient and Portable Natural Language Query Interface for Relational Databases. In *Proc. of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems.* Langhorne, Pa. Gordon and Breach.

Apple Computer, I. 1988. *Hypercard Script Language: the Hypertalk Language.* Reading, Ma.: Addison-Wesley.

Bobrow, R. J., P. Resnick, and R. M. Weischedel. 1990. Multiple Underlying Systems: Translating User Requests into Programs to Produce Answers. In *Proceedings of the 28th Annual Meeting of the ACL*, 227–234. Association for Computational Linguistics.

Cohen, P. 1991. The Role of Natural Language in a Multi-Modal Interface. Technical note, Computer Dialogue Laboratory, SRI International. also in *Proceedings of FRIEND21 International Symposium on Next Generation Human Interface*, Tokyo, Japan, 1991.

Daelemans, W., K. de Smedt, and G. Gazdar. 1992. Inheritance in Natural Language Processing. *Computational Linguistics* 19(2):205–218.

DARPA. 1989–92. *Proc. DARPA Workshops on Speech and Natural Language.* San Mateo: Morgan Kaufmann.

Grosz, B., and C. Sidner. 1986. Attentions, Intentions and the Structure of Discourse. *Computational Linguistics* 12(3):175–204.

Hendrix, G. 1982. Natural Language Interfaces (Panel). *Computational Linguistics* 8(2):55–61.

Iida, M., J. Nerbonne, D. Proudian, and D. Roberts. 1989. Accommodating Complex Applications. In *Proceedings of the First International Workshop on Lexical Acquisition, IJCAI 1989*, ed. U. Zernik. Menlo Park. Morgan Kaufman.

Johnson, T. 1985. *Natural Language Computing: The Commercial Applications.* London: Ovum Ltd.

Kobsa, A., and W. Wahlster. 1988. *Special Issue on User Modeling.* Cambridge, Mass.: MIT Press. Special Issue of *Computational Linguistics*, 14(3).

Martin, P., D. Appelt, and F. Pereira. 1983. Transportability and Generality in a Natural-Language Interface System. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 573–581. Los Altos. IJCAI, Morgan Kaufmann.

Moore, R. C. 1982. Natural Language Access to Databases—Theoretical/Technical Issues. In *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, 44–45.

Nerbonne, J., K. Netter, K. Diagne, L. Dickmann, and J. Klein. 1993. A Diagnostic Tool for German Syntax. *Machine Translation.*

Neumann, G., S. Oepen, and S. P. Spackman. 1993. Design and Implementation of the cosma System. Technical report, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, Germany.

Norman, D. A., and S. W. Draper. 1986. *User-Centered System Design: New Perspectives on Human-Computer Interaction.* Hillsdale, N.J.: Lawrence Erlbaum.

Peddie, J. 1992. *Graphical User Interfaces and Graphic Standards.* New York: McGraw Hill.

Perrault, C. R., and B. J. Grosz. 1988. Natural Language Interfaces. In *Exploring Artificial Intelligence*, ed. H. E. Shrobe, 133–172. San Mateo, CA.: Morgan Kaufmann.

Petrick, S. R. 1976. On Natural Language Based Computer Systems. *IBM Journal of Research and Development* 314–325.

Schank, R., and C. K. Riesbeck (ed.). 1981. *Inside Computer Understanding: Five Programs Plus Miniatures.* Hillsdale, N.J.: Erlbaum.

Searle, J. 1984. *Minds, Brains and Machines.* Cambridge, Mass.: Harvard University Press.

Shneiderman, B. 1980. *Software Psychology.* Winthrop.

Tenant, H., K. Ross, M. Saenz, C. Thompson, and J. Miller. 1983. Menu-Based Natural Language Understanding. In *Proc. of 21st ACL*, 151–158. Cambridge, Mass.

Turing, A. 1963. Computing Machinery and Intelligence. In *Computers and Thought*, ed. E. Feigenbaum and J. Feldman. New York: McGraw-Hill.

Walker, M. 1989. Evaluating Discourse Processing Algorithms. Technical Memo HPL-ISC-TM-89-055, Information Systems Centre, Hewlett-Packard Laboratories.

Walker, M., and S. Whittaker. 1990. Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. In *Proceedings of the 28th ACL Meeting*, 70–78. ACL.

Warren, D. H. 1982. Issues in Natural Language Access to Databases from a Logic Programming Perspective. In *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, 63–66.

Watt, W. C. 1968. Habitability. *American Documentation* 338–351.

Whittaker, S., and P. Stenton. 1989. User Studies and the Design of Natural Language Systems. In *Proceedings of the 4th Annual Meeting of the European Chapter of the Association for Computational Linguistics*, 115–123.