The Articulate Computer

MICHAEL McTEAR

Basil Blackwell
in association with André Deutsch

First published 1987

Basil Blackwell Ltd
108 Cowley Road, Oxford, OX4 1JF, UK

Basil Blackwell Inc.
432 Park Avenue South, Suite 1503
New York, NY 10016, USA

in association with André Deutsch Limited,
105 Great Russell Street,
London WC1B 3LJ, England

British Library Cataloguing in Publication Data

McTear, Michael
The articulate computer.—(The Language library)
1. Linguistics—Data processing
2. Man-machine systems
I. Title
006.3'5 P98
ISBN 0–631–14009–3

Library of Congress Cataloging in Publication Data

McTear, Michael.
The articulate computer.
(Language library)
Bibliography: p.
Includes index.
1. Linguistics—Data processing 2. Pragmatics—Data processing. 3. Artificial intelligence. I. Title.
II. Series.
P98,M33 1987 410'.028'5 86–29950
ISBN 0–631–14009–3

Typeset in 10 on 12pt Sabon
by Cambrian Typesetters, Frimley
Printed in Great Britain by
Billing and Sons Ltd., Worcester

The Articulate Computer

THE LANGUAGE LIBRARY
EDITED BY DAVID CRYSTAL

The Articulate Computer	*Michael McTear*
The Artificial Language Movement	*Andrew Large*
Australian Aboriginal Languages	*Colin Yallop*
Children's Writing and Reading	*Katharine Perera*
A Child's Learning of English	*Paul Fletcher*
A Companion to Old and Middle English Studies	*A. C. Partridge*
A Dictionary of Literary Items (revised)	*J. A. Cuddon*
Early English (revised)	*J. W. Clark*
Early Modern English	*Charles Barber*
English Dialects (revised)	*G. L. Brook*
A Dictionary of Linguistics and Phonetics (second edition)	*David Crystal*
The Foreign-Language Barrier	*J. A. Large*
A History of the English Language	*G. L. Brook*
A History of Foreign-Language Dictionaries	*R. L. Collison*
How Conversation Works	*Ronald Wardhaugh*
An Informal History of the German Language	*W. B. Lockwood*
John Donne: Language and Style	*A. C. Partridge*
Language and Class in Victorian England	*K. C. Phillipps*
The Language of 1984	*W. F. Bolton*
The Language of Walter Scott	*Graham Tulloch*
Language, Society and Identity	*John Edwards*
Languages of the British Isles: Past and Present	*W. B. Lockwood*
Modern Englishes: Pidgins and Creoles	*Loreto Todd*
Non-standard Language in English Literature	*N. F. Blake*
Puns	*Walter Redfern*
Scientific and Technical Translation	*Isadore Pinchuck*
Sense and Sense Development (revised)	*R. A. Waldron*
Speaking of Chinese	*R. Chang & M. Scrogin Chang*
The Study of Dialect	*K. M. Petyt*
A Survey of Structural Linguistics	*G. L. Lepschy*
Thomas Hardy's English	*Ralph W. V. Elliott*
The Treatment of Sounds in Language and Literature	*Raymond Chapman*

Preface

When I first conceived the idea of writing a book about what computers need to 'know' in order to be able to communicate with people in natural languages such as English, I believed that what was required was a model of how humans use language. I also believed that I knew quite a lot about human communication, having previously written a book about the development of conversational abilities in young children. I still hold to my first belief, but my second belief has been shattered. After two years of intensive work I find that there are still so many questions to which I have no answers. But I do not see this as a bad thing – indeed, it is one of the main reasons for attempting to model human intelligence on a computer in the first place, as questions can be brought to light which might not have arisen if we had conducted naturalistic observations or experimental studies.

Natural language processing is an interdisciplinary subject and so writing a book such as this entails making decisions about what to include and what to omit as well as working out what sort of reader to aim at. I have adopted a fairly thematic approach as I wanted to avoid producing a tedious catalogue of the history of natural language processing. I have included most of what I consider to be relevant and interesting, but obviously I had to be selective. I apologize to those whose work has not been covered here and to those whose work has been covered inadequately. As to my audience, I have tried to make the book amenable to a wide range of readers who are interested in how language is used, including linguists, psychologists, computer scientists, cognitive scientists and researchers in artificial intelligence. I assume some prior knowledge of linguistics – roughly what would be covered in an introductory course – but otherwise no technical background should be necessary. I hope I have managed to steer the right course and to avoid being too superficial for some readers and too technical for others.

While writing this book I have benefited immensely from the many

helpful comments and criticisms of friends and colleagues. In particular I would like to mention Richard Caves, Bryan Crow, Alison Henry, Michael Johnson, Karen Maitland and Nigel Shadbolt. I have not always heeded their advice so all the deficiencies which remain are mine. I would like to thank David Crystal for his editorial advice and comments, and my daughter, Siobhan, for her help with the preparation of the diagrams. I am also grateful to Alasdair Gilmore, Chief Technician of the Faculty of Social and Health Sciences at the University of Ulster, who was always ready with his magic touch whenever my communications with the word processor became less than articulate. The copy-editor, Julia Allen, did a fine job and helped to clarify points in the text which may have been unclear.

Finally I would like to acknowledge the unfailing support and encouragement of my wife Sandra, who kept our children at bay while I withdrew for long hours to the attic to work on the book. I would never have finished on time without this support.

Michael McTear

Acknowledgements

Figure 6.1 is reprinted with permission of the publisher from W. G. Lehnert, 1981: A computational theory of human question answering. In A. Joshi, B. Webber and I. Sag (eds), *Elements of discourse understanding*. Cambridge: Cambridge University Press, p. 159. Figure 7.1 is reprinted by permission of the author from B. Grosz, 1978: Discourse. In D. Walker (ed.), *Understanding spoken language*. New York: Elsevier North Holland, p. 254. Copyright © 1978 by Elsevier Science Publishing Co., Inc. Figure 7.2 is reprinted with permission of the publisher from R. Reichman, 1985: *Getting computers to talk like you and me: discourse context, focus, and semantics*. Cambridge, Mass.: MIT Press, p. 26. Figures 8.1 and 8.2 are reprinted with permission of the authors from W. Hoeppner, K. Morik and H. Marburger, 1984: *Talking it over: the natural language dialogue system HAM-ANS*. University of Hamburg research unit for information science and artificial intelligence, report ANS-26, pp. 13, 31. Figure 9.4 is used by permission of the Association for Computational Linguistics from R. M. Weischedel and N. K. Sondheimer, 1983: Meta-rules as a basis for processing ill-formed input. *American Journal of Computational Linguistics*, 3/4, p. 170. Copies of the publication from which this material is derived can be obtained from Dr Donald E. Walker (ACL), Bell Communications Research, 445 South Street, MRE 2A379, Morristown, NJ 07960, USA. Table 8.1 is reprinted with permission of the publisher from E. Rich, 1979: User modelling via stereotypes. *Cognitive Science*, 3 (4), p. 346.

Contents

1

The Articulate Computer: Fact or Fiction?

It is probably true to say that most humans do not know how to communicate with computers. But it is also the case that computers do not know how to communicate with humans. This much will be obvious to any novice who has tried to interact with a computer. However, if more people are to take advantage of the power of modern information technology, it will be essential to develop an easier and more natural mode of communication between humans and machines.

At present the most common way of communicating with a computer is to type in appropriate data and commands at a terminal or on a microcomputer keyboard. This means knowing a computer language, such as BASIC or FORTRAN, which the computer can understand. Other possibilities include the use of a formal query language. Query languages are used for retrieving information from databases. They consist of symbols which are used in prespecified sequences determined by the structure of the database (see example 4 below for an illustration). There are also menu-controlled systems in which the computer may address the user in prepackaged natural language presented in the form of options from which the user can make a choice. Whichever option the user selects elicits particular actions from the computer, such as a display of requested information, the deletion of a file or the printing out of a file.

All of these methods have their disadvantages. Learning to program in a computer language is time-consuming and largely unnecessary as there are specially trained programmers to do such tasks. Imagine having to take a course in car design before being able to drive down the road. Query languages are also difficult to learn and at present there are almost as many different query languages as there are database systems. Menu systems, while being easy to use, restrict what can be communicated to a limited selection from within predetermined choices. In some cases this is acceptable but often it is not.

How useful it would be if we could communicate with computers in our own languages – in English, French, Japanese, Russian or whatever language we normally speak when we talk to people. Speaking is something which comes naturally to most people, whereas learning computer programming languages and formal query language systems does not. Moreover, when using natural language it is possible to have a much more flexible form of communication than can be obtained with any of the previously mentioned methods of human-computer communication.

But is all of this just fantasy? Is it just fiction? Science fiction has made us familiar with computers which use natural language to communicate. Indeed there are many well-known fictional computers ranging from HAL 9000 in Arthur C. Clarke's novel *2001: A Space Odyssey* through Asimov's robots to the Daleks in the popular television series Dr Who. Moreover the idea that computers might be capable of behaving intelligently has fascinated scientists since at least the time of Leibnitz; indeed the notion that intelligent behaviour might be describable in terms of some kind of formal system of calculation can be traced back to the ancient Greeks and the work of Plato. More recently, Alan Turing, regarded by many as the father of modern computer science, wrote in 1950:

> We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried.

Chess has long been regarded as a major challenge in artificial intelligence (AI) and computers have been developed which can play chess at the level of a grandmaster. However chess demands a special type of intelligence and very few people have the ability to play chess well. Other activities, which humans perform with apparent ease, such as recognizing objects visually and producing and understanding language, are precisely those activities which are hardest for computers. Indeed some would argue that programming computers to exhibit intelligent behaviour in these domains is not even possible (Dreyfus 1972; Searle 1980). Humans and computers differ in that humans are adaptable, they can observe patterns, detect relevance and handle unforeseen occurrences and anomalies. Computers, on the other hand, are accurate and fast processors of complex sequential logical operations. In other words, so the argument goes, computers should be used for tasks that are difficult for humans but should not be made to compete with humans in those areas where humans are superior.

Whatever we might feel about these issues, it is the case that considerable progress has been made in recent years in research which involves getting computers to understand and produce natural language. This book will be concerned with what has already been achieved, with what the future holds, and with many of the general and theoretical issues which arise along the way.

Why natural language?

The answer to this question may appear to be obvious, given the need for better communication between people and machines. In fact, research in natural language processing has been motivated by two main aims:

1. to support the construction of natural language interfaces and thus to facilitate communication between humans and computers.
2. to lead to a better understanding of the structure and functions of human language.

The first of these aims is practically motivated with the main goal being the construction of a system that will actually work, whereas the second aim is more theoretical and involves the use of computers as a test-bed of linguistic theory. This separation of aims simplifies the picture somewhat, but in general those whose primary interest is building working systems will seldom incorporate findings from theoretical linguistics or experimental psychology unless they are relevant to the task at hand. Indeed the system might have certain in-built limitations such as the restriction for practical purposes to a small subset of language. For this reason, work on natural language interfaces is often seen as being atheoretical and ad hoc. Nevertheless such work may also stimulate interesting theoretical ideas, and it is being increasingly argued that builders of more intelligent natural language interfaces will need to draw more extensively on theoretical research in natural language processing. We will look at some examples of actual natural language interfaces later. The second aim, on the other hand, is more theoretical and involves the use of computational methods as a means of testing theories about language. This use of computers to test theory has often been referred to as the *computational metaphor*. It requires that the theory be stated explicitly enough to be implementable on a computer and this is taken to be an appropriate way of testing the rigour of the theory.

Natural language processing stems from work in machine translation carried out in the 1950s, when it was hoped that computers would provide rapid and accurate translations of texts, particularly those of interest to readers in the military and scientific communities. The initial euphoria faded as the linguistic problems turned out to be more complex than was originally expected. Since then considerable advances have been made in linguistic theory and there have also been significant developments in the production of working systems which use natural language. AI has become an 'in' discipline and natural language processing is one of its key areas. In part this upsurge in interest was prompted by the announcement in 1979 of a major research programme, the Japanese Fifth Generation project. (The term 'fifth generation' refers to the hardware technology of a computer – for example, first generation computers were built out of thermionic valves, second generation computers out of transistors, and so on.) The Fifth Generation project has as a central aim: the development of a super-intelligent machine capable of storing, manipulating and acquiring information, solving problems by means of sophisticated inference routines, and interacting with human beings through speech, natural language and graphics. The Japanese project elicited a swift response in the USA, Europe and other parts of the world in terms of large-scale and heavily-funded research projects in AI. The Japanese believe that fifth generation computers will bring major benefits to mankind. Some of these benefits will derive from a wider use of knowledge and information processing in all areas of social and economic activity, including a more efficient management of natural resources and the improvement and streamlining of medical information and health care. Natural language will play an important role in such consultation systems as well as in machine translation systems which aim to improve international co-operation in information transfer. For these reasons it is important to know something about the articulate computer and it is hoped that this book will provide some of the answers.

What do we mean by the word 'articulate'? Jean Aitchison wrote an entertaining introduction to psycholinguistics, *The Articulate Mammal* (Aitchison 1983). This title gave me the inspiration for my title. However, there are several differences between Aitchison's book and mine. In her book Aitchison was concerned with questions such as:

Why do we talk?
How do we acquire language?
What happens when we produce or comprehend sentences?

She argued that other questions such as 'How do we talk to other people?' were outside her domain. In other words psycholinguistics was concerned with language processes within the individual rather than with the ways language is used to interact with others. In the present book the meaning of the word 'articulate' will be extended to include those aspects of language which are involved when people communicate with one another. I would argue that this aspect of language is central, as communication is one of the primary functions of language. This does not mean that traditional areas of linguistic enquiry, such as syntax and semantics, are to be excluded. However these areas have already been extensively covered in theoretical and computational linguistics, whereas the areas of discourse and dialogue are less clearly understood. So for this reason too greater emphasis will be put on the investigation of the communicative functions of language rather than on its formal structural characteristics.

One further point needs to be clarified at this stage. When we think of an articulate computer, we might imagine a system that can talk to us and understand what we say when we talk back. In other words speech (in the sense of spoken language) would have to be involved. This book will not be concerned with the question of how computers recognize and produce speech. Instead the simplifying assumption will be made that input to the computer is in typed form.

What is the justification for this approach? There are two responses; one is based on the way research in natural language processing has been conducted while the other is of a more practical nature. As far as the first reason is concerned, until recently there were two quite different traditions in computational linguistics. On the other hand, a considerable amount of attention was devoted to the analysis of speech sounds. This work which involves processing acoustic data in order to represent them in terms of phonetic symbols, and ultimately in words, is often referred to as *speech recognition*. In the main, work on speech recognition has been concerned with the analysis of isolated words or phrases within limited vocabularies of approximately 100 words, with training sessions required for each new speaker. The second approach has taken words as given, usually dealing with input typed into the computer at a terminal, and has been concerned with issues such as the syntactic, semantic and pragmatic analysis of the input. Some recent work has involved the integration of these traditions in the development of large-scale systems capable of understanding connected speech and drawing on multiple sources of knowledge in the process (see, for example, Walker 1978). It is outside the scope of this book to give a detailed account of the technical issues involved in speech recognition though it is clear that future work will need to address further the integration of all linguistic levels in speech understanding systems.

Some examples

At this point it might be helpful to look at some examples of what has been achieved in natural language processing. As mentioned earlier, some systems were developed mainly in order to investigate questions concerned with the nature of human language. The first example, which is taken from the program SHRDLU, is one of the best-known of such systems. Other natural language systems were developed for more practical ends, mainly as interfaces to databases and expert systems. We will present examples of each of these.

SHRDLU: a conversing computer

SHRDLU – a name based on the second half of the 12 most frequent letters in the alphabet – was written in the early 1970s by Winograd (1972) in order to show that many features of natural language conversation could be simulated on a computer. The system operated within a very limited domain, a tiny world containing a box, a table, blocks and pyramids, which could be manipulated according to instructions from the user. In addition, the system could answer questions about its actions and could assimilate new information. The following extract illustrates some of these capabilities. (Note: in this and subsequent examples the convention will be adopted whereby the system's output is printed in upper case and the user's input in lower case, unless user and system are identified by 'U' and 'S'.)

(1)
1 pick up a red block
2 OK
3 grasp the pyramid
4 I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN
5 find a block which is taller than the one you are holding and put it into the box

6 BY 'IT' I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING
7 OK
8 what does the box contain?
9 THE BLUE PYRAMID AND THE BLUE BLOCK

This brief extract illustrates some of SHRDLU's advanced natural language processing abilities. For example: one of the main problems in discourse is locating the objects and persons to which reference is being made, particularly when pronouns and other referring expressions such as definite articles are used. In line 3 SHRDLU was asked to grasp the pyramid, but as there were three pyramids in the scene, it was not obvious which one the user meant. SHRDLU recognized the ambiguity in this instruction and informed the user accordingly. Similarly in line 5 the pronoun *it* could refer either to the block which was being held by SHRDLU or to the taller block which SHRDLU was to find. The system had built in heuristics (rules of thumb) for selecting the most likely alternative and informed the user which choice had been made (just in case it was the wrong one).

Even more impressively, as the next extract shows, the system was able to keep track of previous events which had happened during the interaction with the user and was able to answer questions about why and how it had done certain things.

(2)
1 had you touched any pyramid before you put the green one on the little cube?
2 YES, THE GREEN ONE
3 when did you pick it up?
4 WHILE I WAS STACKING UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE
5 why?
6 TO GET RID OF IT
7 why did you do that?
8 TO CLEAR OFF THE RED CUBE
9 why did you clear off that cube?
10 TO PUT IT ON A LARGE GREEN CUBE

It might be argued that these conversations are very trivial and far removed from what people might normally want to say to computers. It has to be admitted that SHRDLU was extremely limited in its conversational domain, being able only to talk about its tiny world of blocks and pyramids. The system lacked generality – it was unable to discuss anything outside its own world – and even here its knowledge was severely restricted. For example, although the system had knowledge of the geometric specifications of the blocks, it had no knowledge of other attributes such as their weight. Such knowledge would play an important role in a human's planning processes when manipulating large objects. Nevertheless SHRDLU was quite innovatory in comparison with other systems developed in the late sixties and early seventies and it embodied many important principles which have been taken up in later research. Not only that, SHRDLU made people aware of the complexity of the problems which had to be resolved and this has resulted in greater respect for what is involved in the use of language by humans. We will come back to SHRDLU in greater detail in chapter 5.

Natural language interfaces for databases

One of the most widely developed areas in natural language processing is question-answering, where the main application is as a component of an intellectual 'front-end' to a database. An intelligent front-end to a database has two principal functions. First, it relieves the user of the need to be familiar with the structure of the database so that the system at the back-end can produce what the user is requesting in terms of the ways in which it is organized in the database. Second, by providing the facility to interact in the user's natural language, it makes the system more convenient and more flexible. The most usual procedure is that the user's natural language query is translated automatically into a statement in a formal query language which is appropriate to the database. Following this a natural language response is generated on the basis of the retrieved information. One of the most successful early question-answering systems was LUNAR (Woods et al. 1972). LUNAR could answer natural language queries to a database containing information about moon rocks. A more recent system is INTELLECT, which has been used widely as a front-end for commercial databases (Harris 1984). INTELLECT could accept queries such as the following:

(3)
I wonder how actual sales for last month compare to the forecast for people under quota in New England.

This may seem a simple enough question, but it involves a number of complex operations. The system has to print out the names of people under quota in New England, their actual sales for July 1982, their estimated sales for the same month, calculate the change by subtracting the second figure from the first, and return the percentage change by dividing this result by the figure for actual sales and multiplying by 100.