Francis E. Knowles

# DICTIONARIES AND COMPUTERS

## Introduction

Nowadays there is a virtùal consensus that dictionary makers in an academic environment should really be using computers for their professional purposes, just as their counterparts in a commercial setting do, with great skill and success (cf. Goetschalckx and Rolling 1982). No gulf is, incidentally, implied in this contrast - there are no opposing camps, there is no serious information barrier separating these two groups of scholars working in different environments but sharing the same professional skills and dedication. For the growing band of logistically-minded lexicographers friendly attitudes towards computers require neither apologia nor elaboration, but for many who are not committed to main-stream lexicography the suggestion of using computers for one-off dictionary-making projects appears to need justifying. As well as the natural diffidence of those who have not yet acquired hands-on experience of computers - as the jargon has it - there is often a worry that a computer would not be able to provide software and hardware adequate enough to capture, process, display and finally photo-typeset exotic lexi-data without playing a mind-boggling and totally off-putting game of character equivalences en route.

Fortunately, such opinions are founded on misconceptions: lexicographical computing or - for short - lexi-computing can make allowances for missing data, can accommodate highly individualistic approaches, can virtually avoid the hassle of code-conversions. Of course, the cost-effectiveness of computerizing a dictionary-making project, any project, can be assessed and ought to be assessed in each individual case. The real message, however, that ought to be put across to the decreasing number of needlessly timid and hesitant lexicographers is, firstly, that advances in computer science and computer technology have opened up a marvellous range of options for all whose labour of love is devoted to the creation of dictionaries. The technology - or hardware - now offers an extensive and still rapidly growing selection of processors and peripherals. Computer scientists have, for their part, developed an impressive range of software, most of it general-purpose software, but - as the term implies - deployable without too much effort for lexi-computing. In some ways the lexicographer now suffers from an embarras de richesses: he must exercise great care in configuring a computing environment for himself. He should, of course, consult widely but he must first perceive that there are various levels of computer involvement in the general lexicographical process. The lexicographer must therefore decide how the computer ought to be integrated into his personal dictionary-making project.

## Finding a path through the hardware 'jungle'

Let us now take a closer look at the question of, firstly, hardware and, subsequently, software for lexi-computing. Most lexicographers in institutions of higher education work in departments

which are perforce labelled as 'humanities', with the implication that their computing needs are slight and of low priority. Nonetheless, they have access to the central computer service provided by their institutions. In a typical case the facilities available are most likely to be configured as follows. First would come a large computer, or mainframe, supporting many users simultaneously and offering large areas of so-called backing store - available for lexi-data - on both discs and magnetic tapes. This machine would offer both batch-processing of large production jobs on an overnight basis, plus extensive interactive usage for data input, for program editing, and for rapid turn-round of small and not-so-small test jobs. Many peripheral devices would be attached to this machine, notably up to a hundred or so of the now ubiquitous visual display units (VDU's), and line printers providing rough and ready high-volume hardcopy and using a full upper- and lower-case character-set. Further specialized devices could include graph-plotters or even Braille embossers, but the lexicographer would be lucky to find those devices which he needs most. These are optical-character recognition (OCR) equipment for rapid data capture, and at the other end of the processing chain, high-quality printing devices such as so-called letter-quality printers, microfiche printers or photo-composers.

Initial input of data and final output of results are two major hurdles in literary or linguistic computing, notably so in lexi-computing. As far as data input is concerned, it is obviously necessary to distinguish between running text and data already structured in some lexicographically meaningful way. Yet, either way, the data are likely to be copious. OCR offers a tantalizing method of by-passing the keyboarding bottleneck. One particular device, the Kurzweil Data Entry Machine (KDEM), has proven its worth for this purpose.[1] It is possible to train this machine to recognize and interpret during the scanning process not only font styles but, more importantly, most scripts. There are only nine KDEM machines in the U.K. at the present time - by no means all of them in educational institutions - but it is encouraging to be able to say the Oxford University offers a KDEM data-preparation service to external users. The KDEM at the University of Aston recently made it possible to set up a 250,000-word corpus of English journalistic texts within the space of four weeks. There is one lexicographical footnote, with respect to the KDEM, that may be of interest: the machine's software includes the option of a lexicon which is of considerable value in cutting down the operator intervention rate when the print quality of the material being scanned is poor, as is the case with newspaper text. The fact that this lexicon costs the best part of £5,000 makes it very up-market as far as dictionary prices go! Those with no financial worries can purchase a number of foreign-language dictionaries on the same sort of basis. Apart from the KDEM, there are other specialized devices, such as the so-called Cambridge Ideo-matic Encoder, which should be employed for scanning non-alphabetic scripts like Chinese, or for reading hybrid systems like Japanese (cf. Nancarrow 1981).

As far as output is concerned, it is crucial to the lexicographer to be able to produce camera-ready copy straight from the computer he is using for lexi-computing, without any re-keyboarding. The most sophisticated device available in the UK in the academic computing confraternity is the Lasercomp machine at Oxford Univer-

sity (cf. Marriot 1982). This is - as its name strongly suggests - a laser composer and it is fitted out with a vast range of type-styles for an equally vast range of writing systems. Further character-sets can be defined at will. Special pre-processing software is available for setting up correctly the laser commands and for embedding appropriate control codes in the data-stream. Once again, Oxford University performs a noble duty in offering a service - at very reasonable rates - to users from other academic institutions. A few British institutions of higher learning possess photo-typesetters of lesser sophistication than Oxford's Lasercomp but nonetheless capable of being coaxed to produce fully adequate camera-ready copy. With devices such as the KDEM costing about £75,000 and Lasercomp-type devices costing more than twice as much, it is obvious that major facilities such as these need to be purchased by institutions as part of their central facilities, available to all local users and, via a network, to remote users on a regional, national or even international basis.

Having sorted out his initial input and final output problems, the lexicographer will want to make appropriate arrangements for intermediate hardcopy output and for volatile information display via a VDU. Problems arise here because of the need to use facilities not required by the standard user of computers. The lexicographer must often seek to have equipment owned by his institution's computer centre adapted or modified in some way which might make it useless for other types of user. Most computer centres are reluctant, naturally enough, to go down this path but, equally, most would concede that an interface should be provided for the lexicographer to attach his own devices to the system. At this point the ostensibly hapless lexicographer need do no more than choose from the vast range of VDU's and printers commercially available and suitable for his purposes. By way of example - and it is always in some degree invidious to single out an item from a large set: I beg your indulgence for doing this three or four times in the course of this paper - a VDU such as the ICL KDS7362 can be purchased with half-a-dozen toggles to enable different character-sets.[2] The user also has an opportunity to program his own character-sets if he desires. Devices offering such capabilities are now standard market items - the above device costs no more than about £500.

As for printing devices, a machine such as the Anadex WP6000 provides the user with the option of defining character-sets and downloading them into the machine prior to operating it.[3] The manipulation of heavily-mixed language text is entirely under software control. The price of this device, approximately £4,000, is high but justifiable in terms of a serious lexicographical project.

If the lexicographer has pursued his intentions this far he will certainly not be able to sidestep one further tantalizing question. Instead of just interfacing his equipment to a central service for day-to-day working, why not go the whole hog and set up what computer scientists would call a 'dedicated' system? As long as the option exists to network data in and out again, a dedicated system offers the principal advantage that it is a single-user system - if you will permit a tautology - entirely at the beck and call of that single user! Of course, cost is involved, yet for about £6,000 it is possible to acquire and commission a really powerful system such

as a SAGEIV 68000 16-bit microcomputer with an 18-megabyte disc.[4] This sort of configuration gives a processing speed, a storage capacity, and a software environment that is a real competitor to mainframe working.

In this case, though, there is one very important attendant factor to be aware of and that is the discrimination between results and methodology. Some researchers are results-oriented: they want to use computers - often in very sophisticated ways - to produce results they can rely on. Exactly how these results are produced does not concern them too much. Others are fascinated by the ways in which algorithms are developed and programmed and how an operational research environment can be established. For such people (they might be called methodology-oriented) results may actually be of secondary importance. A research team is obliged to encompass both profiles, integrating them as far as possible. In the do-it-yourself mode, the in-house, the dedicated lexi-computing research laboratory there must be a copious supply of methodological expertise, otherwise not much progress is possible. This problem is not negligible even in mainframe mode but it is heightened and exacerbated in the in-house context. The reason for it is directly concerned with the other side of the computing coin: software.

On the alchemy of algorithms

It is possible for a researcher, by using packaged, hermetically-sealed programs designed and implemented by someone else to produce very creditable results. All institutional computing centres mount and maintain so-called applications packages, providing advice on usage and unusual run-time behaviour of such packages. In other words, there is normally an expert systems analyst to turn to for help. Not so in the private set-up: there the onus is on the user to know all he needs to know, to decipher atrociously bad documentation and to fix undocumented bugs. For the lexicographer, moreover, the news from the package front is rather bad: there is no proliferation of lexi-computing packages in widespread general use. Such packages as do exist have not been developed by lexi-computing experts for the exclusive use of lexicographers: Oxford's OCP (cf. Hockey and Marriot 1982) and Birmingham's CLOC (cf. Reed and Schonfelder 1979) can certainly reduce textual data to formats of potential use to lexicographers. UMIST's PTOSYS (cf. Somers and Johnson 1979) is, on the other hand, of great interest to lexicographers, providing an on-line system for tagging the arguments of valency slots, but PTOSYS is not widely known or distributed.

At this point I digress somewhat from my path to discuss the lexicographer who wishes to use the computer as a mere electronic amanuensis. This path is well-trodden and is labelled 'WP' for word-processing; some text-formatting systems provide card-box facilities which might be useful to him, some offer software for detecting and correcting spelling errors, for indexing, even for elementary style monitoring. The automatic dictionaries provided therewith are, however, black boxes and cannot therefore be visually inspected. To conclude this minor digression, let me mention one other commercial product which is potentially very important and which has perhaps a greater intrinsic interest for lexicographers. I refer to the rapidly increasing number of packages being sold for

the purposes of computer-aided instruction (CAI). Many of these packages contain fairly large dictionaries and accompanying sets of vocabulary-based exercises. Some time, however, must yet pass before we are in any position to evaluate the quality and usefulness of such products.

The mention above of card-boxes does, however, introduce a topic which is very, very important for computer-minded lexicographers. It is not really stretching things too much to say that one very special form of package is a data-base management system, commonly abbreviated to DBMS. A DBMS offers the lexicographer what he needs very badly: a dependable system for storing highly-structured information and for retrieving it in a number of different ways. The user can store a chunk or record of information consisting of a main data-field and a number of associated sub-fields. A typical lexicographical example of this would be a definiendum or left-hand side entry from a dictionary as the main data-field in a record with a number of right-hand side elements as sub-fields, such as definiens, pronunciation, etymology, style notes, subject field, grammar code, collocations, etc. Once entered into the system the information relating to a series of records can be retrieved straight or permuted as a result of selecting a particular sub-field as a sort key. A list of entries can then be displayed which coincide in their subject field, for instance. It goes without saying that DBMS's are open-ended: new information can be appended at will. There are many DBMS's available to the users of computers, large and small. My albeit limited experience with the RAPPORT system nonetheless gives me enough confidence to recommend it here[5]; it is a relational data-base management system, well-designed and 'user-friendly'. In addition to its normal mode of operation it can also be accessed from within user programs if need be; the irritant here is that ANSI66 FORTRAN is the required programming language. Even smaller systems such as FAMULUS (cf. Shaw et al. 1974), originally designed as a bibliography information retrieval system, can be very useful and can be adapted and enhanced by programming routines.

The mention of programming languages introduces a major topic of this paper. Every lexicographer, after having used someone else's package for some time, will find himself asking: Why can't this package do this, or that, or the other? This question is the realization that the package's value is exhausted and that any special effects required will have to be programmed. This is the moment of time when the lexicographer either attempts to locate a 'tame' programmer and to give him a complete programming specification of the task in hand - which is a lot more difficult than it sounds - or he takes the decision to try his own hand at programming. Let it be said that as far as lexicographers are concerned the auguries for success are really rather good. Rhetoric aside, any lexicographer could start writing useful programs after as little as twenty-five hours of classroom instruction and private practice in a suitable programming language. The strategic decision to learn how to program is easy, the tactical choice of programming language is fraught with consequences.

Computer programmers often appear to have not just an intellectual, but an emotional, almost visceral commitment to their individual languages and a highly polemical and deprecating attitude to languages they themselves do not use or know. This makes life

difficult for the neophyte attempting to make a rational choice. Truly rational considerations in our instance would include at least the following: is the language supple enough for coding lexi-computing algorithms with a minimum of clumsiness? Does the language obfuscate algorithmic concepts, or does it highlight them? Is the language well-known and widespread? Are there collections of so-called library routines already coded in the language, which might be acquired through contacts with colleagues elsewhere? Does the language run on the hardward available? Is it a general-purpose language or a special-purpose language? After all these questions the time has come to name a few names. Major, widespread languages such as FORTRAN (either in its 66 or 77 incarnation), ALGOL68, PL/I or even COBOL have been and therefore can be used for the purposes of computational linguistics, including lexi-computing. This does not mean, however, that all the answers to the above questions are emphatically affirmative. In the same bracket as these languages is PASCAL, a so-called ALGOL-like language, available on most machines, big, small and tiny (cf. Grogono 1980). PASCAL is an example of a language which is said to be strongly type-checking, that is, it discriminates between and applies a sort of apartheid law to the various types of data-object, such as numbers, strings, logical values, pointers etc. However, PASCAL also exhibits a very positive feature known as modularity, which is one way of saying that programmers using PASCAL are encouraged to break down their global task in a set of separate, indentifiable, testable, transparent modules which interact and eventually integrate to produce the global effects sought. The 'C' language (cf. Kernighan and Ritchie 1978) is from the same stable, so to speak, and appears to be winning over many former PASCAL adherents as a result of the extra facilites it offers. It certainly offers those involved in computational linguistics a number of advantages over PASCAL.

Much work of great value in the field of computational linguistics and Artificial Intelligence (AI) has been programmed in a special-purpose language called LISP (cf. Winston and Horn 1981). LISP's territory is chiefly the USA and its advocates are strong but somewhat dogmatic. However, they now have a real battle on their hands: the powerful competition represented by a logic-programming language called PROLOG (cf. Clocksin and Mellish 1982), popular in Europe, including Eastern Europe; in Poland, for instance, some excellent computational linguistic work has been programmed in PROLOG. More importantly, perhaps, PROLOG has been adopted by the Japanese for their so-called 'Fifth Generation Project' - knowledge programming, intelligent knowledge-based systems (IKBS's), sometimes referred to as expert systems. I make the following remark clearly and with conviction: I cannot think of a better example of a truly expert system than a major, sophisticated, computerized dictionary or - or better - lexical data-base, or LDB. I hope that EURALEX might, as one of its first acts, feel able to take up that point and inject it into the current British debate on Information Technology (IT). Unless this argument makes an impact an important aspects of IT, IKBS's and expert systems will not attract the level of funding they properly deserve in the new climate of research support for these areas. Both LISP and PROLOG are available on most mainframe computers and on many microcomputers, and lexicographers should ask to see these systems demonstrated on some useful examples if they are on the threshold of making a purchase. That is, in fact, a gratuitous and commonplace piece of advice.

Let me conclude this section of my paper with a comment that is admittedly somewhat partial. For about fifteen years computationally-minded linguists have had available a programming language designed almost exclusively for them: SNOBOL (cf. Maurer 1976). In its various versions, such as SNOBOL3, SNOBOL4, FASBOL, MAINBOL, SPITBOL, it has been used to great effect by many linguists attracted by its salient features: a high level of transparency, type-free operation, pliable data-structuring functions, and superb pattern-matching facilities. I venture to suggest, incidentally, that pattern-matching lies at the very basis of all linguistic computing involving primary data such as texts or individual words. SNOBOL is available on very many mainframe and mini-computers. It has made a late entry into the world of micro-computers, having become available only recently; yet an implementation of SNOBOL is to be released for the BBC Model B micro with the so-called second processor in 1984. SNOBOL has been criticized by computer scientists for its lack of modularity, principally attributed to the overtly labelled control structures which are said to frustrate the step-wise refinement of algorithms. SNOBOL's designers took cognizance of this comment, accepting part of its thrust, and came back on the rebound with a language called ICON (cf. Griswold and Griswold 1983). ICON is available free of charge for many computer systems and is eminently portable in its UNIX environment - I am currently engaged in implementing it on a 16-bit micro-computer - and is proving very attractive. In my opinion, it discards everything that is out of date in SNOBOL and retains everything that is good about it. Pattern-matching and modularity have united to form a programming vehicle which I commend to the attention of any lexicographer wishing to acquire or to extend programming skills.

## Research and development

With standard and special-purpose hardware and suitable software tools to hand and equipped with those programming skills which I have stressed as important, what lines of research and development can the lexicographer pursue? One of his first concerns might be to trawl through texts in order to cull lexi-data. May we ignore the obvious but less interesting case of filtering existing files containing already structured lexi-data? The process involved is that of reducing running text forms, each with an actual meaning, to arrays of items, each with a set of potential meanings. Although some of these sets will acquire more than one potential meaning as a result of mappings explicitly suggested by the perusal of an intermediate text concordance, other potential meanings may need to be supplied by the lexicographer from outside his corpus. This might be true even if, say, a 50-million word text corpus were to be used, unless frequency characteristics - carefully evaluated on the basis of a sound experimental design - are themselves used as automatic cut-offs. The lexicographer's first task in normal circumstances, then, is to de-structure text, or to move from a 'textocentric' to a 'lexicocentric' focus. This process yields segments, of course, some of which may be discarded immediately. The decision to discard is a conscious one, relying on some interpretation of the analysis being performed. The easiest case establishes a segmentation process for yielding mere orthographic words, but this easiest case does not really occur all that often! Lemmatization is involved in most instances, and this is a non-trivial task for most languages requiring clever algorithms and mini-dictionaries containing black-

list items. In information science work lexemes semantically and/or derivationally connected but distributed across different parts of speech are often coerced into one form, usually the nominalization of the appropriate verb. Furthermore, items may previously have been tagged automatically - often with the help of an LDB such as the Lancaster-Oslo-Bergen Corpus (cf. Leech et al. 1983). In this case, data pairs, triples or n-tuples are generated and require special handling. Sometimes the items to be selected are discontinuous - verb-cum-particle, for instance. Yet again, the search may be rather more obviously selective in that an item needs to pass through a pattern template to be accepted. Part-of-speech allegiance may, on the other hand, be the selection and grouping criterion. This last example could, probably would, involve heavy parsing with sophisticated disambiguation procedures for resolving the various types of homography.

It is well known to all linguists that the orthographic gaps separating words in texts do not comprise one homogeneous class in terms of their junctural force. Often the juxtaposition or syntagmatic association of words is not merely volatile to the text, that is to say, it is not statistically insignificant. If there is ?rstatistical significance then there is also lexicographical signifi?rcance. In this sort of case the orthographic gap is only masquerading as a gap: what is real is a degree of bonding. So the lexicographer might wish to investigate all multi-word units which are semantically atomic and which exist outside text and prior to text. Fixed idioms are, perhaps, the most obvious example of this phenomenon. It follows that where the syntagmatic relationships are not volatile but are weighted statistically the lexicographer has a task that gets harder as the statistical weighting gets lighter. The computer offers the lexicographer much help in this regard because of the fine control he can exercise by computing the forward predictability of multi-word units immediately such a unit comes under scan. This type of multi-word unit extraction process is, incidentally, fundamental to the mechanics of machine translation (MT) - the lexicographical implication of this is clear: the dictionaries necessary for modelling this mechanism grow by leaps and bounds as a result of the combinatorics of the situation. Yet MT specialists are quite happy to accept the overhead of backtracking heuristics and of an idioms dictionary containing a million entries or more because of its efficiency in choking the otherwise unmanageable number of parsing strategies that need to be tested.

The study of the terminological status of multi-word units in English special-language texts has proved very fruitful; computers were, naturally, used for this purpose. The structure and dynamics of English are a happy accident for the user and an unhappy accident for the dictionary maker who has to list and to describe the static, lexical building blocks of English. To use an analogy, English words nearly always wear civilian clothes - and do not change them according to the company they keep - rather than sporting military uniforms with badges which proclaim their trade and rank, i.e. their part of speech and syntactic function! Thus, English words are gregarious, they associate freely with each other, rarely wedding themselves into compounds, happy in their changing roles. All told, a difficult system to codify lexicographically, very different from German compounds, from French multi-word strings containing function words such as de or à, and from Russian noun phrases with adject

ivized nouns as pre-modifiers. I suggest that only a device such as a computer is really capable of searching for and detecting patterns and rhythms such as these, often deeply buried and not perceptible by visual inspection in a large mass of data. Much of this sort of work is primarily lexicological rather than lexicographical, but I prefer to believe that a useful lexicographical spin-off is possible from every piece of ostensibly lexicological research.

The computer also makes it possible, in a really major way, for the lexicographer to keep all his options open all the time. It is possible to apply different foci to lexical data-bases, once assembled, in a manner which does not pre-empt future projects. In other words, the computer ensures the maximum data potential. Take, for instance, the question of word frequency. This feature of language is not of primary concern or interest to many lexicographers working in a predominantly qualitative mode rather than occupying themselves with the quantitative aspects of their data. I ought nonetheless to insert here the comment that desirable lexicographical features such as controlled defining vocabularies and circularity-stifling mechanisms have a largely statistical basis. It cannot be denied that the computer's capabilities are such that it requires almost a wilful act of abandon on the part of researchers to ignore the statistical behaviour of lexis. Statistical criteria in lexicography have proven their worth almost across the board from the optimization of foreign language learning to the partitioning of special registers and the characterization of functional styles. It should, moreover, not be forgotten that computerized lexical data-bases can themselves be fine-tuned and maximized for efficiency by in-built modules of a statistical nature. Of course, it is likewise a simple matter to monitor the use of a computerized dictionary - and dictionary use is now clearly recognized to be an important concern for the makers and publishers of dictionaries.

Once a computerized lexical data-base has been fully established and kitted out with appropriate software, the question arises as to the multiplicity of ways in which the resident information can be manipulated and displayed. The obvious point is that many different dictionaries - some permanent, some ephemeral - can be generated from a single LDB. Useful data sub-sets are easily created by suppressing certain entries or certain information fields and by putting the other main or subsidiary fields into the prominence of the sorting and listing slot, i.e. the left-hand side of the dictionary. This type of approach is a topic of major interest and potential in computational lexicography. I hope that enough has been said to indicate that the lexical data-base philosophy is powerful and flexible enough to encompass everything from the major, highly-structured, explanatory or bi/multilingual dictionary, replete with systematic information, at the top end, via the hierarchically and tightly-structured information-science thesaurus or the semantic dictionary of items arrayed according to their distinctive feature configurations - in the middle, somewhere - to the lowly crossword-solver's companion, denuded of all information apart from its listing structure.

I should now like to mention two similar pieces of lexi-computing research, one carried out in the USSR and the other in the USA. The Soviet work was done by Karaulov who set out to compute an automatic thesaurus, or semantic dictionary, of Russian (cf. Karaulov

1981). The basic strategy involved was to analyze the right-hand side definitions - which can be usefully viewed as microtexts - in two major Russian explanatory dictionaries, to filter out function words, to truncate derivational morphemes and use the resulting sets of conflated descriptors in order to establish degrees of connexity between left-hand side definienda. Analogous techniques are used, incidentally, in text linguistics and information science for profiling documents and for constructing automatic text summaries or abstracts. Karaulov's work attempts to elicit a set of so-called semantic factors on this sort of basis, which can then be arrayed - after some cyclical processing and additional data input - as a conceptual hierarchy. The above-mentioned cyclical processing involved a frequency analysis and ordering of the semantic factors within individual definitions so as to yield a subsequent cut-off threshold. The final product contains an array of some 1600 descriptors, each with its list of semantic factors and with its definiens tabulating associated words, cross-referenced and keyed into the semantic factors. When the next and final stage of Karaulov's system is complete it will contain three types of entry point: (1) 'conceptual' - from descriptor to word, (2) 'lexemic' - from word to descriptor, and (3) 'isosemic' - from semantic factor, or seme, to word. The whole LDB will provide Russian speakers and students of Russian linguistics with a unique data-base to catalyze lexicographic lateral thinking.

The analogous American work I referred to above is Amsler's (1980) analysis of the MERRIAM-WEBSTER POCKET DICTIONARY. Amsler wanted to determine whether useful semantic information could be derived, with computer assistance, from dictionary definitions (some 45,000 altogether), information which might enable a complete taxonomy of the dictionary's entries to be elaborated. Amsler's findings were many and varied, but all of them important and suggestive of further work which needs to be done. They included, for instance, a clear and substantiated statement that dictionary definitions can be used for the componential analysis of case-frame argument patterns. Amsler also shows that taxonomies can be created computationally with a minimum of human assistance over disambiguating definition genus terms. The MERRIAM-WEBSTER yielded no less than 27,000 nodes for nouns and 12,000 for verbs. There is also the interesting and potentially important account of what might be called the iso-grammar of definition statements which gives such prominence to the genus term that automatic indentification of the genus is virtually guaranteed. Neither Amsler's nor Karaulov's work would have been possible without computers, and both investigations focus on the search for conceptual and formulatory systems of which we are, perhaps, only subliminally aware but which provide much food for thought for those interested in devising novel methods for the presentation of language data in lexicographical form.

Not merely a means to an end!

Computers also offer the lexicographer new help, or at least present him with challenges in other respects. This conference bears eloquent witness to the professionalism which inspires lexicographers to design and bring to fruition dictionaries pro bono publico. Paramount is the need to provide reference tools which explain meaning and content, but much attention is now rightly being devoted to finding ways of enhancing the value of dictionaries by

optimizing the ways of presenting information in them. In other words, the form also has its content.

In a world experiencing a rapid evolution in the spectrum of dictionary profiles - each profile emerging to satisfy a perceived need - the idea of the language-for-special-purposes (or LSP) dictionary is gaining acceptance as a valid dictionary type. This brand of dictionary - hybrid of dictionary proper and encyclopaedia - is designed for adults who are not experts in some particular field but who wish to be. In contradistinction to compendia for fully-fledged experts - which trainees often find obstructive and obfuscating - the LSP dictionary has to accommodate to certain pedagogical requirements by structuring its form in such a way as to embody more content, if I may put it like that. This dictionary must contain as many potential learning paths as possible; some gateways into the dictionary maze need to be open, some need to be shut. A degree of scrambling is called for in the network which models the perceived knowledge of the content, structure and linkages comprising the particular subject discipline involved. The lexical items which denote the greatest degree of conceptual detail - the analogy is that of the greatest indentation in a subject thesaurus - are not directly but only indirectly accessible via superior levels of the conceptual hierarchy and the dictionary's user must 'take a walk' through the various nodes to get to his target concept. Such would be the top-down operational mode - bottom-up would be starting on the floor of the hierarchy and gradually filling in the broader term slots until a full contextualization of the subject discipline had been achieved.

I think it follows that the only device readily capable of implementing a lexicographical CAI system of this sort is a computer. Scrambled textbooks were never really viable on account of their users' irritation over the problems of handling them - darting backwards and forwards through the pages at the behest of a clue on each page! I wish to investigate this in a pilot way during a SERC-funded research project I am about to commence which will assess the potential for visual look-up computer dictionaries vis-à-vis traditional hand-held volumes. Advances in micro-miniaturization mean that really sophisticated vest-pocket, calculator-type - even customized - dictionaries are going to be available in the near future. In fact, the first such devices are already on sale.[6] The growth of IT is so rapid and is gaining such acceptance that the general public is going to be surprised, or rather disappointed, if usefully large lexical data-bases are not readily available on cheap terms. Schoolchildren will take these devices and systems for granted and secretaries in electronic offices will need bug-free computerized bilingual dictionaries and other document-processing aids for the much more realistic machine-aided translation (MAT) systems which are now just beginning to emerge.

One type of computerized visual look-up system which has already established itself is, of course, the so-called term bank. Very large and impressive term banks - which are a special form of LDB - are to be found in a number of countries in which govermental interest, encouragement, and financial support have made it possible to develop, usually within the context of standards organizations, major facilities to optimize the work of teminologists and to propagandize, if I may put it like that, their results. Technical

translators and technical writers are able to consult these term banks on a dial-up basis, thereby bypassing the necessity of possessing a complete range of technical translating or defining dictionaries. I believe that a number of valuable spin-offs from term bank systems and usage will affect and benefit a much wider community of language users and processors in the near future (cf. Sager and McNaught 1981; Rondeau 1981).

Computational lexicography for machine and machine-aided translation systems is a subject which I have dealt with elsewhere (cf. Knowles 1982, 1983) and I cannot, for lack of space, treat it adequately here. I can, however, and do underline the basic identity of approach which MT/MAT specialists share with their more orthodox fellow-lexicographers involved in mono/bi/multilingual dictionary-making for the purposes of HT: human translation. The same need for meticulousness and consistency is there, the need for quasi-complete coverage is more pressing. The major difference is the vital need for a sophisticated calculus for the semantic representation of all the various types of meaning. The success of an MT/MAT system depends crucially on the standard of lexicographical expertise and thoroughness invested in it. MT/MAT systems designers have basically accepted that a poor LDB can frustrate the entire effort of the system it is part of and MT/MAT lexicographers are themselves mostly fully aware of the need for a fully integrated systems approach.

There is a fair measure of agreement on what actual modules are needed for a serious MT system: a general-language frequency dictionary, a whole cascade of special-register glossaries, an extensive idioms dictionary, a dictionary of pseudo-idioms (i.e. constantly recurring fixed phrases with incomplete semantic content), a main analysis dictionary of stems, a transfer dictionary, and a generation dictionary. Entries in the master-stem dictionary need to be strictly co-ordinated and to be replete with morpho-syntactic, semantic, functional-stylistic, and thesaural information. Furthermore, all these various modules need to be fully cross-referenced and keyed in to each other. Part of this cross-referencing process could well involve establishing the analysis dictionary as a reversible mechanism so as to yield a generation dictionary. The visual analogy is, of course, not really appropriate, but it is as if the dictionary can be entered from both the left-hand and from the right-hand side. Each definiens can double as a definiendum and vice versa. The usual one-to-many mapping characteristic of the transition from definiendum to definiens is avoided by the invocation of computable test routines on the contextual meaning representation achieved so far. The result of such tests gives a green light for one, and only one equivalence. Setting up all these complex linkages inside the LDB is not a once-and-for-all activity: various personnel and operational requirements have to be satisfied. It is easy to corrupt an LDB by allowing unvetted data to seep into it. Lexicographers must always be on front-line duty to wrestle with the so-called not-found words during translation processing. They can, as post-editors, always lexically fix a translation but they must keep careful records so that additions to the LDB are fully systematized and controlled. This less hectic process comprises, of course, the major part of their dictionary-coding work: fortunately, the computer itself can help them enormously in this task by menu-driven operations and even by suggesting routine parameter values.

## Conclusion

Let me now conclude by returning to a matter referred to above. Too many gullible customers are being tempted to buy software products in the form of dictionaries or vocabulary-based CAI routines, which have been designed and implemented without the expert advice or active involvement of lexicographers. I submit that lexicographers cannot stand on the sidelines and be unaffected by these developments. Professional pride alone should be the guarantee of that, but it must be a matter of some slight concern that the professional reputation of dictionary makers might be sullied - language teachers and translators know the feeling of having no quarantine only too well! - and tarred by association with the vendors of commercially available software products masquerading as dictionaries when they do not deserve the name and when in reality they are - if I may use a nonce-word which I have used before in a different context - nothing more than 'lexi-con-tricks'!

## Notes

1 The Kurzweil Data Entry Machine (KDEM) system and documentation are distributed in the U.K. by Omnifont International Ltd., 12 High Street, Chalfont St. Giles HP8 4QA.

2 Functions and Operating Instructions for the ICL KDS7362 Character Display are provided by ICL Doc. No. 212-0027, 1982.

3 The Anadex WP6000 system and documentation are distributed in the U.K. by Anadex Ltd., Weaver House, Station Road, Hook, Basingstoke RG27 9JY.

4 The SAGEIV 16-bit Microcomputer system and documentation are distributed in the U.K. by T.D.I. Ltd., 29 Alma Vale Rd., Bristol.

5 Cf. Deen (1979), Date (1981) and the RAPPORT manuals published by Logica in 1979 and 1980.

6 The LANGENSCHEIDT ALPHA 8 electronic dictionary, for example, is a calculator-type 8,000-word German-English dictionary, using genuine lexicographical principles (see report in Der Spiegel, No. 4/1983).

## References

Amsler, R.A. (1980) The Structure of the MERRIAM-WEBSTER POCKET DICTIONARY. Ph.D. thesis (Report TR-164 Department of Computer Science and Linguistic Research Center), Austin: University of Texas

Clocksin, W.F. and Mellish, C.S. (1982) Programming in PROLOG. Berlin: Springer

Date, C.J. (1981) An Introduction to Database Systems. Reading, Massachusetts: Addison-Wesley

Deen, S.M. (1979) Fundamentals of Data Base Systems. London: Macmillan

Goetschalckx, J. and Rolling L. eds. (1982) Lexicography in the Electronic Age. Amsterdam: North-Holland

Griswold, R.E. and Griswold, M.T. (1983) The ICON Programming Language. Englewood Cliffs, New Jersey: Prentice-Hall

Grogono, P. (1980) Programming in PASCAL. Reading, Massachusetts: Addison-Wesley

Hockey, S. and Marriot, I. (1982) Oxford Concordance Program. Oxford University Computing Service

Karaulov, Ju.N. (1981) Lingvističeskoe konstruirovanie i tezaurus literaturnogo jazyka. Moscow: Nauka

Kernighan, B.W. and Ritchie, D.M. (1978) The 'C' Programming Language. Reading, Massachusetts: Addison-Wesley

Knowles, F.E. (1982) "The pivotal role of the various dictionaries in an MT system" in Practical Experience of Machine Translation ed. by V. Lawson. Amsterdam: North-Holland

Knowles, F.E. (1983) "Towards the machine dictionary: 'mechanical' dictionaries" in Lexicography: Principles and Practice ed. by R.R.K. Hartmann. London-New York: Academic Press

Leech, G. et al. (1983) "The automatic tagging of the LOB Corpus" ICAME News 7: 13-33

Marriot, I. (1982) LASERCHECK User's Manual. Oxford University Computing Service

Maurer, W.D. (1976) The Programmer's Introduction to SNOBOL. Amsterdam: Elsevier

Nancarrow, P.H. (1981) "A brief account of the development and first major installation of the Ideo-matic Chinese character encoder" ALLC Bulletin 8, 3: 263-265

Reed, A. and Schonfelder, J.L. (1979) "CLOC: a general-purpose concordance and collocations generator" in Advances in Computer-aided Literary and Linguistic Research ed. by D.E. Ager et al. Birmingham: AMLC

Rondeau, G. (1981) Introduction à la terminologie. Montréal: CEC

Sager, J.C. and McNaught, J. (1981) Selected Survey of Linguistic Data Banks in Europe. Manchester: BL R&D and CCL/UMIST

Shaw, A. et al. (1974) FAMULUS. UCL Computer Centre

Somers, H.L. and Johnson, R.L. (1979) "PTOSYS: an interactive system for 'understanding' texts using a dynamic strategy for creating and updating dictionary entries" in The Analysis of Meaning ed. by M. MacCafferty and K. Gray. London: ASLIB

Winston, P. and Horn, B.K.P. (1981) LISP. Reading, Massachusetts: Addison-Wesley