Creating a Historical Dictionary of Hungarian with the Aid of Computer

Júlia Pajzs

Summary: in 1984 the Hungarian Academy of Sciences decided to create an unabridged monolingual dictionary of Hungarian based on historical principles. Both collecting the source material and compiling the dictionary will be done with the aid of the computer. As a first step about 13 million running words will be keyboarded and the dictionary entries will be written using the concordances generated from the running texts. This paper gives an outline of the project from the computational point of view.

In 1984 the Hungarian Academy of Sciences decided to create an unabridged monolingual dictionary based on historical principles. Although there were already a huge number of dictionary slips for that purpose it was decided that the whole collection would start all over again by using a computer for both collecting and editing. The new concept of dictionary making was very similar to that of the Trésor de la langue française. Instead of gleaning the interesting quotations from a large amount of text, the source will be provided by recording running text on the computer. From the running texts flexible concordances will be generated and the lexicographers will compile the dictionary entries by using these concordances. As there is not much experience in making historical dictionaries by computer we had to design a purpose-built collecting and editing system for it. However, we tried to use the ideas of the Dictionary of Old English and the Trésor de la langue Française as much as possible.

First of all we had to decide how to select the source material of our dictionary. The idea was to collect about 10-15 million words from several running texts in order to get a representative sample of the Hungarian vocabulary. Taking into consideration the quantity of the printed materials in the different centuries, the number of words to collect changes from century to century. From the 16th-18th centuries we plan to collect about 4-5 million words, and from the 19th-20th centuries about 7-9 million words will be keyboarded. Since the selected texts should represent the vocabulary of every given century, we collect the texts from several authors including less famous ones. We usually type just some pages continuously, so we rarely type complete books (e.g. an entire novel) (Short stories and poems are of course complete.) In some special cases we type entire works when they are epoch-making (e.g. the classic narrative poem *Toldi* by János ARANY, a 19th century poet).

We always work from printed publications. It was also to be decided which edition of a given work should be used. It was finally agreed that the critical editions should be used if possible. In other cases we use the edition which was published in the author's lifetime and seems the most reliable.

After posing these basic principles historians of the literature of the different centuries selected the appropriate sample texts. We shall also collect some words from the technical literature of the early centuries, as selected by the historians of the different trades.

As soon as a decision was reached as to the amount and proportion of the running words to be collected and when the sample texts for recording were chosen, the actual keyboarding began. We are typing the running texts as accurately as possible by using a very simple coding system. We represent the special Hungarian characters and the historical characters by combination of letters and numbers. We use some codes to mark the end of line, end of stanza, end of paragraph etc. Every sample text is recorded on a separate file, all of the files have a bibliographic code.

There is a corresponding bibliographic file where all the necessary data are recorded, including the data of keyboarding, proofreading and correcting. This is kept in a DBASE III database file, and there is a program for updating and retrieving different kind of data from it.

In order to get concordances of the lexemes we decided to develop a morphological analyser program. To be able to appreciate the difficulties of the automatic lemmatisation let me mention just a few features of the Hungarian morphology: (i) a great number of inflected forms (for example a noun can have about 760 different forms), (ii) polysemy or polyfunction of the same ending, (iii) a great number of compound words, (iv) separating verbal suffixes. Although we are aware that there are (already) numerous sophisticated ways of morphological analysis, we wanted to use something quick, efficient and good enough for lexicographic purposes. Our analyser program has to be able to find the boundary of the lexeme and suffixes even when the actual form of the root differs from the lexeme, and (of course) to identify all the suffixes and the lexemes. With this aim in mind we decided to use a lexeme and a suffix database.

The lexeme database contains about 70 thousand lexemes coded for part of speech and homonyms, if any. When the root can occur in different forms in the running words the database also contains the possible variants and the lexeme. The variant roots are generated by a special program which uses the codes of the Dictionary of Hungarian Inflection (this dictionary was compiled by László Elekfi). The variants are not kept in their full form but in a special coded way.

In the suffix database all the suffixes are kept with their alphanumeric codes. The reverse form of the suffixes are also kept there and the database is indexed on the reverse suffix.

Using these two databases the program first tries to find the longest matching root, then it cuts off the root from the running word and searches for the remaining string in the suffix database. When there is more than one suffix in the remaining string, the program always chooses the longest suffix which can be matched from the right, and again searches for the remaining string. At each step the algorithm checks if the endings can follow each other in that order. At the last step when no unanalysed string remained the root and the suffix codes are checked to see if they are possible combinations. Checking is made difficult by the fact that after cutting of the rightmost suffix it is impossible to decide whether or not this suffix can follow the given root. This is so because a root may be followed by a suffix that changes the part of speech of the derived root compared to the original root. For instance, when analysing the word *csinál+ó+k+nak* 'for (the) makers', the ambigous morpheme *-nak* (1. dative; 2. 3rd plural suffix) cannot be analysed as a verbal root (*csinál*) plus *-nak* (3rd plural). (See Fig. 1) To arrive at a correct analysis the program must identify the intervening morphemes, (i.e. *-ó* 'er' and *-k* 'plural'), and then decide that

-nak is a dative following the derived nominal root *csinálók-*, rather than a verbal suffix after the verbal root *csinál-*.

On the other hand, this method of analysis provides an easy way of handling the compounds: when the remaining string between the root and the suffixes cannot be analysed as a suffix it can be a part of a compound. In this case the program searches for the remaining string in the lexeme database. The prefixes are handled in the same way as the compounds: these are kept in the lexeme database with their code, the program cuts them off at the beginning of the analysis and it finds the root after identifying the endings. Some of the compounds and derivatives are kept in the lexeme database; in these cases they are analysed in the same way as the simple lexemes.

csinál+ó + k + nak
Original keyboarded text:
'make' 'er' 'plural' dative
(for the makers)

csinál+ok
'make' '1st singular'
(I make)

csinál+nak
'make' 3rd plural
(they make)

Figure 1

Nem mondhatom el senkinek,...
(I can't tell anyone)
Analysed text:
nem⟨MO⟩ mond⟨IG+⟩hat⟨HAT⟩
om⟨Tel⟩
el⟨IK⟩ senki⟨FN⟩nek⟨DAT⟩...

Figure 2

By using this program most of the lemmatization can be solved automatically: the user only has to correct the output when the program either cannot find any good division or finds more than one. In both cases it writes a special sign after the running word in question. Sometimes it means that the user has not only to correct the lemmatized text file but also to add some new lexemes or variants to the lexicon. A sample from the original and the lemmatized text file can be seen in Fig 2. There is a grammatical tag after each morpheme indicating the type of the morpheme. The tags of the lexemes contain the code of the part of speech and code of the homonym if there are any. When the lexeme is a part of a derivative there is a '+' sign in the tag. In case of the suffixes the tag consists of the code of the suffix.

As soon as a sufficient amount of text is already keyboarded and analysed morphologically, the text files will be copied into one large-scale file one after the other. An indexing program will build a tree on this major text file, the tree will contain the initial position of each morpheme and its code. Another index will be built on the beginning and end of parts of texts. A flexible concordance program will use these index-trees.

In the age of optical storage the full-text concordances with invariable context length seem to be outdated. The full concordance of the 13 million words would need in all about 26 million lines (presupposing concordances with two line-length) which would be a waste of space even on microfiche. On the other hand, a concordance on microfiche is not really easy to handle and cannot be ordered in different

ways. So instead of creating a huge but impracticable concordance we would like to use a very sophisticated software for recall our text database. The main functions of this software will be:

— search for the context of one word
— search for the context of two or more cooccuring words
— search for two or more cooccuring codes (one can search for examples of a syntactic pattern in this way)
— combining the above functions (search for only one kind of syntactic context of a given word, or a list of words etc.)
In all of the above cases the output of the program must be flexible in various aspects:
— length of the context
— order of the concordance (alphabetized either on the right or the left hand context or in chronological order)
— amount of the output (the program must be able to choose a certain amount of examples either in a random way or according to a special algorithm: for example one occurrence from every author, etc.)
— selection of output according to style and/or age.

Another program will provide different kind of statistics and word lists, for example: frequency list, reverse word list etc.

Special work-stations will be used for editing the entries. A window system will help the lexicographers to write the entries and to choose the best quotations from the text databank. (See Fig. 3) In the main window a scheme of the entry will appear which will be filled in by the lexicographer during the session. In the second large window the user will be able to run the flexible concordance program described above. He should also be able to mark the quotations if they might be interesting or not, and to pick up the most meaningful ones and copy them into the dictionary entry. In a third window one might want to check the already compiled dictionary entries in order to write them in a similar way or to avoid cyclical meaning definitions etc.

címszó: a1ll
(entry)
alakváltozatok:
(variants)
szófaj 1:
(part of speech 1)
jelentés 1:
(meaning definition 1.)
idézetek 1.:
(qoutations) .....

címszó: u21
alakváltozatok:
szófaj 1.: IG
jelentés 1.:
idézetek 1.:

locus code page text	key word	year
1630434554 184 ..embernek abban	a1ll, hogy meg vallya..	1525
1739322376 265 ../Ki tudja hol	a1ll meg, mintha látn..	1653
1883621878 13 ..a e1s fokozata	a1ll elo3tte, melyek..	1752

Figure 3

At the moment our main task to record the running texts and the lexeme database including the inflection codes. In the meantime we are developing the analyser program, which is tested by a sample lexeme database and a complete suffix database. (The sample lexeme database contains about 6000 lexeme taken from the Frequency Dictionary of Hungarian, without their inflection codes and their variants, therefore the program is not yet able to analyse the variant roots.) We would like to begin the analysis of the real texts as soon as the lexeme database is complete. We plan to finish keyboarding in about two or three years, in the meantime we hope to get a suitable hardware for running the concordance programme and the editing programme for the dictionary.

References

Amos, A. 1984. 'Computers and Lexicography: The Dictionary of Old English. Status Report on the DOE Project.' Unpublished manuscript. University of Toronto.

Gonnet, G.H. 1987. 'PAT — An Efficient Text Searching System. University of Waterloo Centre for the New OED.' Unpublished manuscript.

Martin, E. 1984. 'Une banque de données sur la langue française' in *BRISES Bulletin de recherches sur l'information en sciences économiques, humaines et sociales.* Avril 1984. No. 4.

Venezky, R. 1987. 'Unseen Users, Unknown Systems. Computer Design for a Scholar's Dictionary' in *Proceedings of the Third Annual Conference of the UW Centre for the New OED.* University of Waterloo Centre for the New OED. 113—119.