[*Proceedings of the National Symposium on Machine Translation*, UCLA February 1960]

Session 7: THE DICTIONARY

GLOSSARY LOOKUP MADE EASY

Hugh Kelly
Ted Ziehe

The RAND Corporation

Summary

Most of the work on the dictionary problem for machine translation has consisted of attempts to reduce the amount of information involved, thus bringing the problem within the capabilities of currently available or soon-to-be-available computing equipment. This paper presents a technique for handling the problem with currently available computing equipment and without the complexities of information compression. In essence, the approach is to compile a glossary of forms from the current text and then to retrieve information about each from the dictionary as the information is needed in the translation process.

The general role of the dictionary in machine translation is well understood. Its main function is to store information about the source language which the computer will need when translating text. Accompanying the dictionary must be a method for retrieving information that pertains to each word in text. The method must provide complete and accurate retrieval at a rate compatible with the rest of the translation process. More specific questions about dictionary organization and contents are not so clearly understood. We hope in the following to clarify some problems of dictionary organization that have resulted from attempts to implement dictionary operations on digital computers.

To simplify the description we assume that one machine system (a high-speed computer) does dictionary lookup as a part of translation. The operations used for storage and retrieval must be defined for all words in the language, but whether the information file itself resembles anything the linguist recognizes as a dictionary, word list, glossary, etc., is immaterial. The important point is that the linguist should be able to enter modifications and retrieve information in a format with which he is familiar. In short, the machine should work for the linguist, not the reverse.

Most students of the dictionary problem in machine translation have assumed that only random access to dictionary entries can meet the needs of a machine translation system. Attempts to implement dictionary operations based on this assumption have led to the conclusions either that computer storage capacity must be increased

[1, 2] or that new linguistic techniques for handling dictionary entries must be developed [3,4] . Thus, MT has provided a share of the motivation for developing higher-capacity computer stores. Also, a great deal of linguistic work has gone into studies of morphology, in an effort to reduce the storage requirements that the dictionary system places on the computer.

The many linguistic techniques for reducing the amount of dictionary information that have been proposed all organize the dictionary's contents around prefixes, stems, suffixes, etc. A significant reduction in the volume of store information is thus realized, especially for a highly inflected language such as Russian. For English the reduction in size is less striking. This approach requires that: (1) each text word be separated into smaller elements to establish a correspondence between the occurrence and dictionary entries, and (2) the information retrieved from several entries in the dictionary be synthesized into a description of the particular word. The logical scheme used to accomplish the former influences the placement of information in the dictionary file. Implementation of the latter requires storage of information needed only for synthesis.

We suggest the application of certain data-processing techniques as a solution to the problem. But first, we must define two terms so that their meaning will be clearly understood:

1. form --any unique sequence of alphabetic characters that can appear in a language preceded and followed by a space.
2. occurrence -- an instance of a form in text.

We propose a method for selecting only dictionary information required by the text being translated and a means for passing the information directly to the occurrences in text. We accomplish this by compiling a list of text forms as text is read by the computer. A random-storage scheme, based on the spelling of forms, provides an economical way to compile this text-form list. Dictionary forms found to match forms in the text list are marked. A location in the computer store is also named for each marked form; dictionary information about the form stored at this location can be retrieved directly by occurrences of the form in text. Finally, information is retrieved from the dictionary as required by stages of the translation

process--the grammatic description for sentence-structure determination, equivalent-choice information for semantic analysis, and target-language equivalents for output construction.

The dictionary is a form dictionary, at least in the sense that complete forms are used as the basis for matching text occurrences with dictionary entries. Also, the dictionary is divided into at least two parts: the list of dictionary forms and the file of information that pertains to these forms. A more detailed description of dictionary operations--text lookup and dictionary modification--give a clearer picture.

Text lookup, as we will describe it, consists of three steps. The first is compiling a list of text forms, assigning an information cell to each, and replacing text occurrences with the information cell assigned to the form of each occurrence. For this step the computer memory is separated into three regions: cells in the W-region are used for storage of the forms in the text-form list; cells in the X-region and Y-region are reserved as information cells for text forms.

When an occurrence 0_i is isolated during text reading, a random memory address X_i, the address of a cell in the X-region, is computed from the form of 0_i. Let $F(0_i)$ denote the form of 0_i. If cell X_i has not previously been assigned as the information cell of a form in the text-form list, it is now assigned as the information cell of $F(0_i)$. The form itself is stored in the next available cells of the W-region, beginning in cell $W_{i'}$ The address $W_{j'}$ and the number of cells required to store the form are written in X_i; the information cell X_i is saved to represent the text occurrence. Text reading continues with the next occurrence.

Let us assume that $F(0_i)$ is identical to the form of an occurrence 0_j which preceded 0_i in the text. When this situation exists, the address X_i will equal X_j which was produced from $F(0_j)$. If X_j was assigned as the information cell for $F(0_j)$, the routine can detect that $F(0_i)$ is identical to $F(0_j)$ by comparing $F(0_i)$ with the form stored at location $W_{j'}$. The address $W_{j'}$ is stored in the cell X_j. When, as in this case, the two forms match, the address X_j is saved to represent the occurrence 0_i. Text reading continues with the next occurrence.

A third situation is possible. The formula for computing random addresses from the form of each occurrence will not give a distinct address for each distinct form. Thus, when more than one distinct form leads to a particular cell in the X-region, a chain of information cells must be created to accommodate the forms, one cell in the chain for each form. If $F(O_i)$ leads to an address X_i that is equal to the address computed from $F(O_j)$, even though $F(O_i)$ does not match $F(O_j)$, the chain of information cells is extended from X_j by storing the address of the next available cell in the Y-region, $Y_{i''}$, in X_j . The cell $Y_{i''}$ becomes the second information cell in the chain and is assigned as the information cell of $F(O_i)$. A third cell can be added by storing the address of another Y-cell in $Y_{i''}$; similarly, as many cells are added as are required. Each information cell in the chain contains the address of the W-cell where the form to which it is assigned is stored. Each cell except the last in the chain also contains the address of the Y-cell that is the next element of the chain; the absence of such a link in the last cell indicates the end of the chain. Hence, when the address X_i is computed from $F(O_i)$, the cell X_i and all Y-cells in its chain must be inspected to determine whether $F(O_i)$ is already in the form list or whether it should be added to the form list and the chain. When the information cell for O_i has been determined, it is saved as a representation of O_i . Text reading continues with the next occurrence.

Text reading is terminated when a pre-determined number of forms have been stored in the text-form list. This initiates the second step of glossary lookup--connecting the information cell of forms in the text-form list to dictionary forms. Each form represented by the dictionary is looked up in the text-form list. Each time a dictionary form matches a text form, the information cell of the matching text form is saved. The number of dictionary forms skipped since the last one matched is also saved. These two pieces of information for each dictionary form that is matched by a text form constitute the table of dictionary usage. If each text form is marked when matched with a dictionary form, the text forms not contained in the dictionary can be identified when all dictionary forms have been read. The appropriate action for handling these forms can be taken

at that time.

Each dictionary form is looked up in the text-form list by the same method used to look up a new text occurrence in the form list during text reading. A random address X_i that lies within the X-region of memory mentioned earlier is computed from the i-th dictionary form. If cell X_i is an information cell, it and any information cells in the Y-region that have been linked to X_i each contain an address in the W-region where a potentially matching form is stored. The dictionary form is compared with each of these text forms. When a match is found, an entry is made in the table of dictionary usage. If cell X_i is not an information cell we conclude that the i-th dictionary form is not in the text list.

These two steps essentially complete the lookup operation. The final step merely uses the table of dictionary usage to select the dictionary information that pertains to each form matched in the text-form list, and uses the list of information cells recorded in text order to attach the appropriate information to each occurrence in text. The list of text forms in the W-region of memory and the contents of the information cells in the X and Y-regions are no longer required. Only the assignment of the information cells is important.

The first stage of translation after glossary lookup is structural analysis of the input text. The grammatical description of each occurrence in the text must be retrieved from the dictionary to permit such an analysis. A description of this process will serve to illustrate how any type of information can be retrieved from the dictionary and attached to each text occurrence.

The grammatic descriptions of all forms in the dictionary are recorded in a separate part of the dictionary file. The order is identical to the ordering of the forms they describe. When entries are being retrieved from this file, the table of dictionary usage indicates which entries to skip and which entries to store in the computer. This selection-rejection process takes place as the file is read. Each entry that is selected for storage is written into the next available cells of the W-region. The address of the first cell and the number of cells used is written in the information cell for the form. (The address of the information cell is also supplied by the table of dictionary usage.) When the complete file has been read, the

grammatic descriptions for all text forms found in the dictionary have been stored in the W-region; the information cell assigned to each text form contains the address of the grammatic description of the form it represents. Hence, the description of each text occurrence can be retrieved by reading the list of text-ordered information-cell addresses and outputting the description indicated by the information cell for each occurrence.

The only requirements on dictionary information made by the text-lookup operation are that each form represented by the dictionary be available for lookup in the text-form list and that information for each form be available in a sequence identical with the sequence of the forms. This leaves the ordering of entries variable. (Here an entry is a form plus the information that pertains to it.)

Two very useful ways for modifying a form-dictionary are the addition to the dictionary of complete paradigms rather than single forms and the application of a single change to more than one dictionary form. The former is intended to decrease the amount of work necessary to extend dictionary coverage. The latter is useful for modifying information about some or all forms of a word, hence reducing the work required to improve dictionary contents. Applying the techniques developed at Harvard [5] for generating a paradigm from a representative form and its classification, we can add all forms of a word to the dictionary at once. An extension of the principle would permit entering a grammatic description of each form. Equivalents could be assigned to the paradigm either at the time it is added to the dictionary or after the word has been studied in context. Thus, one can think of a dictionary entry as a word rather than a form.

If all forms of a paradigm are grouped together within the dictionary, a considerable reduction in the amount of information required is possible. For example, the inflected forms of a word can be represented, insofar as regular inflection allows, by a stem and a set of endings to be attached. (Indeed, the set of endings can be replaced by the name of a set of endings.) The full forms can be derived from such information just prior to the lookup of the form in the text-form list. Similarly, if the equivalents for the forms of a word do not vary, the equivalents need be entered only once with

an indication that they apply to each form. The dictionary system is in no way dependent upon such summarization or designed around it. When irregularity and variation prevent summarizing, information is written in complete detail. Entries are summarized only when by doing so the amount of information retained in the dictionary is reduced and the time required for dictionary operations is decreased.

Dictionary printing can be simplified if the word-entries within the dictionary are ordered alphabetically on a representative form of each word. Then an alphabetical list of representative forms with representative information for each word can readily be made. A detailed printing of all information for specific entries would be made only on request.

Separation of dictionary information into parts, each part corresponding to a stage in the translation process, presents no difficulties for entry correction. Either all corrections of a particular type of information are collected and made at once, or all types of information for an entry are made available to the correction program concurrently.

The dictionary system just described is characterized by the following features:

(a) A limited amount of text is read before translation of the text is completed.

(b) Text is not alphabetized by the computer, but rather randomized.

(c) The limit on the amount of text read at once varies as a function of memory size and the degree of factoring desirable in the translation process.

(d) The complete dictionary is read once each time the limit on text is reached.

(e) The only limit on dictionary size is processing time. (Processing time increases linearly with dictionary size; the logic is in no way affected by dictionary size.)

(f) There is no searching of the dictionary at any time; a directed search is made in the text-form list when the randomizing scheme fails to produce a distinct address for distinct forms.

(g) The system is defined for, and makes no exception of, any form in the language.

(h) Information input to the dictionary and printed from it conforms with standard linguistic formats.

A 32, 000-word computer store can handle the dictionary information for at least 7, 000 forms. Statistics from over 200, 000 running words of physics text in Russian show that between 30, 000 and 60, 000 words of text yield 7, 000 unique forms. Hence, with a memory of this size translation can proceed using blocks of text over 30, 000 words long. This is roughly equivalent to 125 pages of printed text.

The amount of dictionary information that will eventually be required for each form can only be estimated. If one assumes that the average form length equals 12 characters, the average grammatic description equals 12 characters, and the average amount of semantic choice information is 26 characters, then one has a total of 50 characters per form. With this amount of information and a 62, 500-character-per-second tape-reading rate, the IBM 7090 requires about 3 minutes to read a dictionary of 200, 000 forms. From this we estimate that dictionary lookup can be performed for 30, 000 occurrences in a 50, 000-form dictionary in less than 3 minutes; in a 200, 000-form dictionary in less than 7 minutes.

The quality of the formula for randomizing entries in the text-word list will determine the number of one-form chains and the length of multi-form chains. This becomes an important consideration only if the rate of operation becomes bound by internal processing rather than tape read-time. A formula that will yield 80% unique addresses for a 30, 000-word text over & 10, 000-address range appears to be quite feasible.

Communications between the dictionary and the lexicographer can be in a format familiar to the linguist. Entries in the dictionary consist of a completely inflected word, but only the basic information must be supplied: stem, ending set, basic grammatic description, affix-determined variations for the basic grammatic description, and the equivalents. Dictionary corrections are based primarily on the word-group but information about each individual form is available for change. Listings of the dictionary can be made in complete detail on each form. In these ways the human effort involved in assembling and maintaining the dictionary is minimized.

This dictionary system, therefore, makes possible high-

speed lookup operations using a high-capacity dictionary. The dictionary's format is flexible; the type and amount of information stored have no affect on the logic that is used. The system results from the application of techniques which allow one economically to use the basic simplicity of form matching. The emphasis is on retention of enough information in the dictionary to obviate development of new linguistic procedures.

REFERENCES

[1] King, Gilbert W. and Irving L. Wieselman, "Stochastic Methods of Machine Translation", International Telemeter Corporation, (1956).

[2] Wall, R. E., Jr., "Some of the Engineering Aspects of the Machine Translation of Language", Papers Presented at the AIEE Summer and Pacific General Meeting, (1956).

[3] Lamb, Sidney J. and William H. Jacobsen, Jr., "A High-Speed Large-Capacity Dictionary System", University of California, Berkeley, (1959).

[4] Pacak, Milos, "Morphology in Terms of Mechanical Translation", Preprints of Papers for an International Conference for Standards on a Common Language for Machine Searching and Translation, Western Reserve University and Rand Development Corporation, Cleveland, Ohio (1959).

[5] Oettinger, A. G., W. Foust, V. Giuliano, K. Magassy, and L. Matejka, "Linguistic and Machine Methods for Compiling and Updating the Harvard Automatic Dictionary", Preprints of Papers for the International Conference on Scientific Information, National Research Council, Washington, D. C., Part V, pp 137-160, (1958).