三重県立看護大学紀要，2,67～72. 1998.

A Computerized Dictionary Lookup and Construction System for the Japanese Reader of English

Myles O' Brien*1 Masao Tamaru*2 Iori Nagaya*2
Ai Kawagoe*2 Takashi Osawa*2

【Abstract】 A system was written in Macintosh HyperCard to assist Japanese readers of English by allowing them to look up the Japanese translation of English words by a single mouse click. Any EDICT format dictionary file, or combination of these, can be loaded into the program and utilized. The text to be read may be pasted into the program or read from a file. Users can also construct their own EDICT format dictionary files. The program provides flexibility and ease of use, at the expense of speed and memory requirement.

【Key words】 Electronic dictionary, English－Japanese translation, EDICT format

1. Introduction

Background

Reading text in another language requires a knowledge of both the vocabulary and grammar of that language. For complete understanding, knowledge of idiom, slang, culture, history, etc. may be needed, depending on the material being read. For this reason, satisfactory machine translation has proven to be incomparably more difficult to attain than had been anticipated by the pioneers in the field. I have discussed this point in more detail elsewhere.[1] In fact, until the explosive growth in the availability and power of personal computers within the present decade, there was no form of electronic translation assistance available to the general reader; words still had to be looked up manually in printed dictionaries, as decades or centuries earlier. Faster personal computers with vastly increased disk and memory capacity enabled complete dictionaries to be installed, affording the user greater ease and speed in looking up words. A step up in automation is provided by what may be called annotation software. This scans the original text and shows translations for all but the simpler words, either writing them above or below the undisturbed original, or inserting them into the text in place of the original words. In either case, the word order of the original is unchanged and the work of translating the overall meaning is left to the user. Fully automatic translation programs which carry out the complete process are also available and, while very far from being perfect, are good enough to be of some practical use in many cases. At present, many new personal computers in Japan come with an impressive array of such software as

*1 Myles O' BRIEN : Mie Prefectural College of Nursing
*2 Masao TAMARU, Iori NAGAYA, Ai KAWAGOE, Takashi OSAWA : Suzuka Medical Technical College

part of the pre-installed package. This may include dictionaries (for Japanese only and for translation to or from English), bilingual concordancing, and annotation or translation software. In less than ten years the situation has changed out of all recognition. Furthermore, if the computer is connected to the internet, a much larger range of specialized dictionaries and more powerful annotation and translation software is available.

Although the usefulness of the above developments for the Japanese reader of English is undeniable, effortless reading and understanding, or translation into Japanese, of any text in English is far from being realized. While a limited range of material in English can be translated by computer into Japanese which is understandable (even if a little strange) and accurately reflects the original, in many cases the results are at best unreliable, and at worst totally nonsensical. Human vigilance and intervention remain essential in using translation software effectively. An obvious unavoidable irony is that those most capable of proper intervention are those who have the greatest translation skills, and hence the least need for the software, while those who need it most are those least capable of using it effectively. However, for a skilled translator it can save some typing, and for a less skilled user it can be helpful if caution is exercised.

As a general rule, for reasons mentioned at the start, the more factual or scientific the English, the better the results; the more conversational or literary, the worse. Scientific papers are supposed to be written as clearly, concisely, and unambiguously as possible; so they are most amenable to computer translation. Literature, even if it is translated *accurately,* cannot be translated *well* by a machine. Whether this is a fundamental fact, or merely a temporary situation, is a most fascinating and controversial point, but well beyond the scope of the present discussion.

The need for a new program

Assuming the Japanese reader has a reasonably good knowledge of English, one of the biggest barriers to reading about specialized topics is the relevant specialized vocabulary. Although there are many specialized printed dictionaries, the electronic dictionaries available commercially are almost entirely for general use. However, the internet has been rapidly filling this gap in recent years. A notable example is the Life Science Dictionary (LSD) Project[2] of Kyoto University which is sponsored by the Telecommunications Advancement Foundation of Japan. This project has assembled a dictionary database of life science terms, currently over 30,000, which is freely available for download or online consultation. Over the internet, even an entire English document can be pasted into a box and submitted, returning in a matter of seconds extensively annotated in Japanese. Annotation is offered in two forms: Japanese translations may be inserted into the text, replacing English words, or the original may be left undisturbed except for a change in the color of words for which a translation is available; clicking one of these words will focus on its Japanese translation in a box underneath the text. This box contains a list of all the translated words with their Japanese equivalents, so it can be a very useful resource if the user wants to assemble a specialized vocabulary list. A selection of other dictionary databases can be chosen instead of, or in addition to, the LSD vocabulary.

The LSD dictionary is one of many Japanese-English dictionaries in the EDICT format, named after the original general-use dictionary file compiled by Jim Breen of Monash University in Australia.[3] The EDICT file has grown to over 60,000 entries since its inception in 1991 as a voluntary project. It is freely available for non-commercial use, and is employed by many freeware dictionary and wordprocessing programs. The format of each entry is: KANJI [KANA]/ english_1/ english_2/ ……/, or in the case of a word which is only in kana: KANA/ english_1/ english_2/……/. Thus, each Japanese word can have as many alternative English definitions as desired. This format makes it a little easier to use in Japanese-to-English rather than English-to-Japanese applications, but it is used by both.

Other specialized dictionaries in EDICT format include COMPDIC[4] (terms used in the computing and telecommunications industries), LAWDIC[5] (legal terminology), FINDIC[6] (financial terms), and GEODIC[7] (geology). These may be consulted over the internet, or downloaded and used by local software. On the internet, centralized access makes the situation simpler. For instance, all of the dictionary files mentioned above can be consulted through the WWWJDIC server at Monash,[8] though local access may be preferable, depending on the situation. If it is necessary, or preferred, to work offline, freeware programs are available for various platforms. For the Macintosh, the premier program is MacJdic[9] from Dan Crevier of Harvard. This program provides a fast lookup function for words in English or Japanese, returning translations from the EDICT file very quickly even when many relevant entries are found. To facilitate speed it generates an index file based on the version of the EDICT file on the user's computer. If an updated version of EDICT is downloaded a new index file needs to be generated. While this is satisfactory if the user wants to consult only EDICT, if it is required to consult other EDICT format dictionaries, or combinations thereof, a stratagem of renaming (also appending for multiple dictionaries) and re-indexing must be employed. This is not alone troublesome and potentially confusing, but care must be taken not to break the rules governing the use of the dictionaries. While the rules concerning EDICT are very relaxed (provided the use is non-commercial), it is forbidden, for instance, to distribute altered copies of the LSD file. Thus it was decided that a new Macintosh program which avoided these problems should be useful.

2. Design, Construction, and Operation

Design considerations and objectives

The primary consideration was to facilitate consultation of any combination of available EDICT format dictionaries for Japanese translations of English words while reading a document. A secondary consideration, also not provided by MacJdic, was to enable users to construct personal EDICT format dictionaries to supplement those already available. Also, MacJdic runs as a separate program, so that if the document being read is in a window of another application, MacJdic must be activated to look up a word, following which the previous application must be brought to the front again. This process is helped by the fact that the contents of the clipboard are automatically pasted into MacJdic's lookup window when it is activated, but it is still relatively cumbersome.

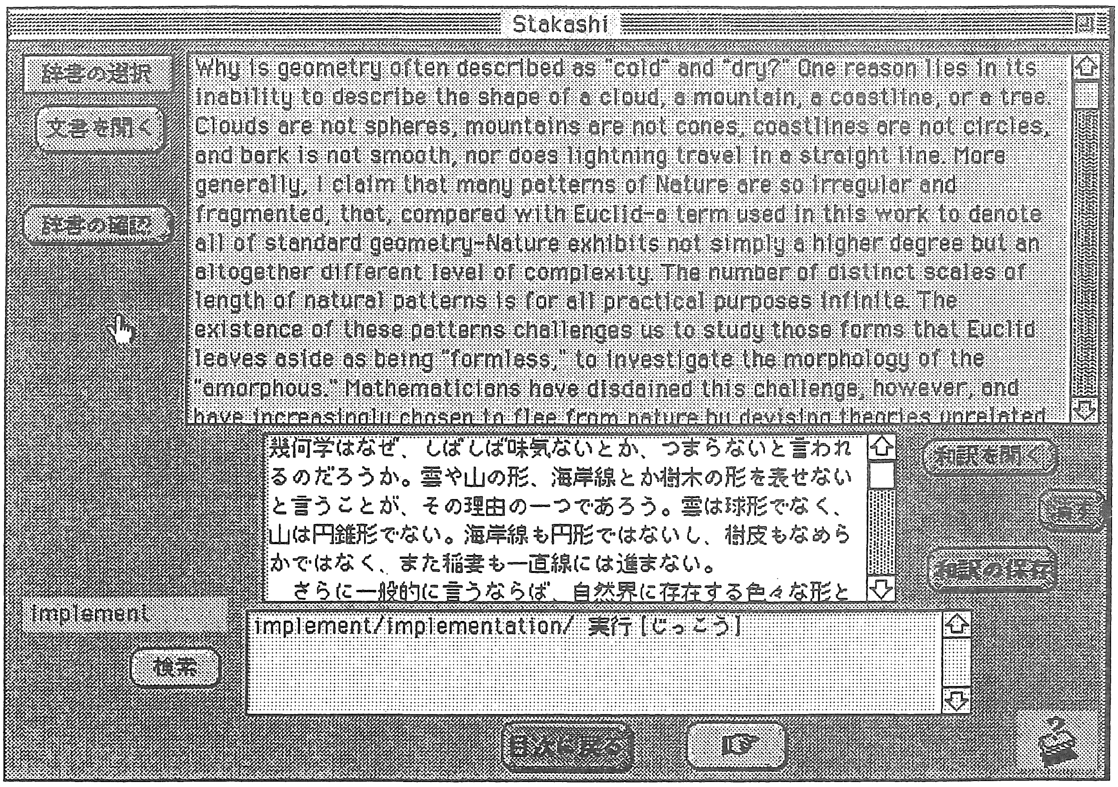
It was decided that, in the new program, when the document is available as a text file, simply clicking on a word should provide its translation. A box would also be provided into which words could be typed or pasted for lookup.

The work was to be undertaken by three fourth-year students of Medical Information Science as their graduation research project. Therefore Macintosh HyperCard was chosen as the development tool, being sufficiently powerful to attain the objectives and also sufficiently easy for relatively inexperienced programmers to learn and use. A drawback of HyperCard is its lack of speed, and this was anticipated as a problem in looking up large dictionary files.

However, third-party software solutions can often circumvent this difficulty, and it was decided to proceed.

Construction and basic operation

The heart of the program is the lookup system for searching the dictionary files, so this was investigated first. It was decided to attempt to implement it without the use of any indexing system. This would decrease speed and increase memory requirements, but would make the system easier for the user to maintain, there being no need to generate, re-generate, and keep track of index files. This could be an especially troublesome and error-prone task in this system where it was hoped to provide ease of use of multiple dictionary files, including those constructed by the user. A fully automated index maintenance system would add greatly to the work of development and the complexity (and hence susceptibility to error) of the system. Thus it was decided to read dictionary files into memory where a relatively fast search could be carried out by HyperCard's built-in search function. The program was constructed so that on startup the user is prompted to select the dictionary files to use, the default option being those used when it was previously run. These are then written into text resources, one for each dictionary file, using the freeware external command (XCMD) WriteText.[10] The simpler option of using HyperCard fields is not practical because of their size limit of approximately 35,000 characters per field. The size of text resources is limited only by available memory. The algorithm to search for a word was written as a user-defined function which takes the word as its argument. It reads the text resources containing the dictionary files contiguously into memory, searches for all lines containing the word, and returns a list of the Japanese headword and the relevant English from each line. Thus, a search for "blood", for instance will return not just "blood" and its translation, but items such as "blood group", "blood pressure", etc. This worked satisfactorily with small trial dictionary files but proved impr-



Why is geometry often described as "cold" and "dry?" One reason lies in its inability to describe the shape of a cloud, a mountain, a coastline, or a tree. Clouds are not spheres, mountains are not cones, coastlines are not circles, and bark is not smooth, nor does lightning travel in a straight line. More generally, I claim that many patterns of Nature are so irregular and fragmented, that, compared with Euclid-a term used in this work to denote all of standard geometry-Nature exhibits not simply a higher degree but an altogether different level of complexity. The number of distinct scales of length of natural patterns is for all practical purposes infinite. The existence of these patterns challenges us to study those forms that Euclid leaves aside as being "formless," to investigate the morphology of the "amorphous." Mathematicians have disdained this challenge, however, and have increasingly chosen to flee from nature by devising theories unrelated

幾何学はなぜ、しばしば味気ないとか、つまらないと言われるのだろうか。雲や山の形、海岸線とか樹木の形を表せないと言うことが、その理由の一つであろう。雲は球形でなく、山は円錐形でない。海岸線も円形ではないし、樹皮もなめらかではなく、また稲妻も一直線には進まない。
　さらに一般的に言うならば、自然界に存在する色々な形と

implement

implement/implementation/ 実行 [じっこう]

acticably slow with bigger files: a search for a word with only one occurrence in the LSD file, for example, took over 30 seconds on a Macintosh Quadra with a 33 MHz 68040 processor. Using a 66MHz Power Macintosh gave no significant improvement. It was therefore decided to compile the searching function into an external function (XFCN) using the third-party software CompileIt from Heizer Software.[11] This allows compilation of HyperTalk code, which is interpreted and runs slowly, into much faster executable code. A dramatic improvement in speed was obtained, the time required for the search mentioned above being reduced to about two seconds. This is not particularly fast, and increases correspondingly if there is more than one occurrence of the word. However, it was thought reasonable, considering the use for which the application was intended; it could be expected that the typical user would spend considerable time thinking over the meaning of the text being read, so a delay of a few seconds in searching for an individual word would be relatively insignificant..

The illustration shows the main card of the finished program. The big field at the top is that containing the text to be read or translated. This can be pasted in, or read from a file using the "文書を開く" button on the left. The buttons above and below this allow the user to select dictionary files, and check which are currently selected. Clicking on a word in the top field will display translation information in the field at the bottom. The middle field is for the user to type a translation of the text while working, if desired. This saves switching between applications if the basic text formatting available in a HyperCard field is sufficient. The contents of the field can be saved as a file and recalled later using the buttons on the right. Words can be typed directly into the small field on the bottom left, and clicking on the "検索" button underneath will bring up the translation information in the same way as for a word clicked in the main text. This is useful for independently looking up words not occurring at all in the text, or for trying an alternative form of a word if the form in the text is thought not to be the root form. Most EDICT format files include only the English root form, and to cope with this the program automatically truncates the ending and searches again for any word not found, if it ends in "s" or "ed". This simple measure covers most cases, especially as helping with technical terms rather than highly inflecting words is the primary aim of the program, and other cases are left up to the user. If the word is not found in the currently loaded dictionary files, the user is offered a choice of adding it and its translation (which the user must find elsewhere) to a user's file. Another card comes to the front for this process. The user can also add freely to this file at any time, of course.. Detailed explanations of all the program's functions are available through the help icon in the bottom right-hand corner.

3. Evaluation and Conclusions

The system succeeds in its primary objective of enabling the one-click lookup of words in several EDICT format dictionary files simultaneously, while not altering the original files in any way. It is also fairly easy to operate and maintain. Furthermore, the facility to make personal tailor-made EDICT format dictionaries could be very useful for specialists. On the other hand, it has weak points in the areas of speed, memory requirements, and sophistication:

The problem of speed has been mentioned already. It is not impracticably slow, but is certainly very slow compared to what would be expected of a commercial dictionary program these days. As regards its memory requirements, about 10 megabytes of memory need to be allocated to HyperCard for it to work properly with the LSD dictionary alone. This will not be a problem on a typical new machine, but it may make it difficult or impossible to run on those with less RAM. As regards sophistication, it cannot be denied that the system compares poorly with the facilities now available on the internet. The LSD project service referred to above which returns a complete word list or a fully annotated document, rather than requiring a word by word check by the reader, is much more user-friendly. Such a feature could be added to our system without too much programming difficulty, but the slow speed would become a real problem with documents of any substantial length. However, for working without an internet connection, or using dictionary files not available on internet services, the system can still be of use. If an improved version were to be developed, the problems of speed and sophistication would need to be addressed.

注

1) O'Brien M: Intelligent Computer Assisted Language Learning, 鈴鹿医療科学技術大学紀要, 1, 167-174, 1994.

2) http://lsd.pharm.kyoto-u.ac.jp/

3) http://www.dgs.monash.edu.au/~jwb/edict.html

4) ftp://ftp.monash.edu.au/pub/nihongo/compdic.doc

5) ftp://ftp.monash.edu.au/pub/nihongo/lawgldoc.euc

6) ftp://ftp.monash.edu.au/pub/nihongo/findic.doc

7) ftp://ftp.monash.edu.au/pub/nihongo/geodic.doc

8) http://www.dgs.monash.edu.au/~jwb/wwwjdic.html

9) http://www.boingo.com/dan/

10) http://www.seanet.com/~jonpugh/software/TextRez.sit.hqx

11) http://www.royalsoftware.com/descriptions/compileit.html