The Integration of MT and MAT

Robert Freuderking, Dean Grannen, Peter Coussen, Sergei Nirenburg
Center for Machine Translation
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA

Abstract

Although automatic machine translation (MT) is beyond the state of the art today, the need for increased translator productivity is urgent. We describe the PANGLOSS system, which integrates MT with machine-aided translation (MAT), with particular emphasis on the CMAT (Component Machine-Aided Translation) editor. The current MT system generally translates units of the source text smaller than full sentences. The gaps left between such units are translated by lookup in several online resource sources. The result is a string of target language words and phrases we call components. The CMAT editor is designed to help the user quickly and easily assemble this fragmentary output from the MT system into a high-quality text. The user interface, its implementation, and its relation to the rest of the PANGLOSS system are described.

1. Introduction

Fully automated machine translation of unconstrained texts is beyond the state of the art today. The need for mechanizing the translation process is, however, very urgent and pressing. It is desirable, therefore, to seek ways of both speeding up the process of translating texts and making it less expensive. In this paper we describe an environment that facilitates the integration of automatic machine translation (MT) and machine-aided translation (MAT).

This environment, called the Translator’s Workstation (TWS) [Nirenburg 1992], has been developed in the framework of the PANGLOSS machine translation project. The main goal of this project is to develop a system that will, from the very beginning, produce high-quality output. This can only be attained currently by keeping the human being in the translation loop. The main measure of progress in the development of the PANGLOSS system is the gradual increase in the level of automation. Currently PANGLOSS translates from Spanish into English, although additional source languages are planned.

The analyzer used in this configuration of PANGLOSS is a version of the ULTRA Spanish analyzer from NMSU [Farwell 1990], while generation is carried out by the PENMAN generator from ISI [Mann 1983]. The Translator’s Workstation provides the user interface and the integration platform. It is similar in spirit to systems such as the Translator’s Workbench [Kagler 1991].

The processing in PANGLOSS goes as follows:

1. an input passage is broken into sentences;
2. a fully-automated translation of each full sentence is attempted; if it fails, then
3. a fully-automated translation of smaller chunks of text is attempted (in the first PANGLOSS configuration, PANGLOSS MARK 1, these were noun phrases);
4. the material that does not get covered by noun phrases is treated in a “word-for-word” mode, whereby translation suggestions for each word (or phrase) are sought in the system’s MT lexicons, an online bilingual dictionary, and a set of user-supplied glossaries;
5. the resulting list of translated noun phrases and translation suggestions for words and phrases is displayed in a special editor window, where the human user finalizes the translation.

The entire process can thus be viewed as helping a human translator by doing parts of the job automatically and making the rest less time-consuming. We have designed and implemented an intelligent post-editor to assist the user in this task, the CMAT (Component Machine-Aided Translation) editor. It allows the user to select, move,
and delete words and phrases (components) quickly and easily, using dynamically-changing menus.

Since the CMAT editor is part of the TWS, it should be easily adaptable to use with changing underlying MT technology, as is the TWS as a whole.

2. The User's View

Suppose the user selects a region of source text, as shown in figure 1, and submits it to be machine-translated. The result appears in the target window, as shown in figure 2.

- The label region, which contains the word or phrase in the source text that produced this particular component.
- The function region, which contains the post-editing Move, Delete, and Finish functions. When the user selects Move, the component disappears, and the mouse pointer changes shape, indicating that a Move is in progress. The component is reinserted into the text at the nearest word break to the point where the user clicks the mouse again. Delete simply deletes the component. Finish removes the component markers, indicating that CMAT editing for this component is finished.
- The alternative region, which is of varying length, contains alternative translations of the source word or phrase. The source word or phrase is also present as an alternative, when available, as translators may wish to leave some source language words temporarily in the target text, and return to them later. Selecting one of the alternatives replaces the original selection for this component with the alternative, while the latter becomes an alternative in the alternative region.

Using these popup menus, the user can move, replace, or delete an output component with one or two mouse actions, rapidly turning the string of translated words and phrases into a coherent, high-quality target language text (figure 4). Note that the user is not forced to use the CMAT editor at any particular time. Its use can be intermingled with other translation activities, according to the user's preferences.

3. Implementation

The TWS is implemented in Common LISP. It communicates through CLM (the Common LISP-Motif interface) (Bäcker 1992) to use Motif widgets inside of the X11 window system.

We use X11 and Motif because they provide a standard, coherent, multi-window system for user interface, and were the only system available off-the-shelf at the time we began work on TWS. Unfortunately, they are written in C, and our research is based on Common LISP. CLM solves this problem, by providing the connection between Common LISP and the Motif system.

3.1. The internal sentence representation

When a sentence arrives at the CMAT editor, it consists of a list of components, which together cover all of the words in the source sentence (figure 5). These components are of three types:

1. MT-generated strings. Phrases translated by the MT system are represented simply as the generated target language string.

2. Glossary entries. Phrases not translated by the MT system, but found in the user glossaries, are each represented by a component list, a list containing the source string (source language phrase), the identifier (GLOSS), and a glossary entry list; a list of the possible target language phrases corresponding to the source language phrase.

3. Dictionary entries. Words not covered by either of the above are represented by a component list containing the source string, the identifier (MT) and a target language string list: a list of the corresponding target language words found in the MT system's lexicon; and finally the identifier (DICTIONARY) and a dictionary entry list: a list of target language words found in the machine-readable dictionary.

3.2. Representing components within the CMAT editor

MT-generated strings are directly output to the target language window, and are not further processed by the CMAT system. The processing of the other two types of components proceeds as follows. The CMAT editor uses a knowledge base and a working memory, each implemented with a hash table. The knowledge base stores static information for a component's menu, while the working memory provides a mapping between the knowledge base and the components currently present in the target buffer. This separation is necessary because any given component generally occurs often in a given text, but there is only one menu associated with a particular component.

Knowledge base structures are indexed by their component source strings. These structures contain four elements, one slot for each of the three possible translation lists in a component list plus a fourth slot containing the candidate list. This list is a union of the first three lists, with the elements' positions varying to reflect current estimates of their likelihood of being chosen by the user. Initially, the items from the target language string list appear first in the list and glossary entries appear second.
Figure 5: Component list

Initial preference is given to items in the target language list because these items are more likely to be the correct translations of the source string in a given domain.

When a component list is passed to the CMAT editor to be displayed, the CMAT editor first checks to see if a structure for the component already exists in the knowledge base. If an entry does not exist, one is created. Then the first component is chosen from the candidate list, and this component is displayed with brackets in the editor window. In the working memory, a pointer to the knowledge base entry is stored, indexed by the displayed component.

When the user clicks the mouse within a CMAT component, LISIP cannot directly access the clicked item, due to the separation between C and LISIP discussed at the beginning of this section. Rather, LISIP must search backward and forward for the delimiting brackets, and then request a copy of the string between them from CLM. Once it gets that string, it uses it as the index into the working memory, and from there gets the index into the knowledge base. The list of alternative translations for the component can then be obtained from the knowledge base structure.

When a component is moved in the editor window, nothing changes in the internal representation of the CMAT editor. When a component is deleted, the pointer in the working memory is removed. When an alternative translation is chosen from the candidate list, the old component is replaced with a new component in the CMAT editor. The pointer in the working memory is removed from its old location and stored under the new component. Also the candidate list in the knowledge base structure is updated. The new candidate is moved to the front of the list as the most likely candidate. The changing of the order of the candidate list in the knowledge base structure is done so that the more recently chosen components can be selected with a minimum amount of visual searching and mouse motion.

The component markers can be removed from the editor window with a single mouse action or keystroke. When this is done, the pointers in the working memory are removed, but the entries in the knowledge base remain. These are retained in order to remember the user's preferences, in the frequent case where future translations contain these components. When the user finishes a session with the CMAT editor, the entries in the knowledge base will contain a summary of the user's interactions with each component list. This summary can be saved as a file for later analysis, for example to make decisions about modifying the order of glossary and dictionary entries. This file can also be loaded into the knowledge base the next time the user uses the system.

4. Conclusion and Future Work

The CMAT editor we have described here, in conjunction with fragmentary MT and word-for-word translation, allows the translator to produce high-quality translations much more quickly and easily than the simple combination of a text editor and an online dictionary (results of an initial experiment to demonstrate this appear in [Frederking 1993]). It will remain a crucial module in PANGLOS until the MT system reaches the point of translating full sentences on a regular basis.

The CMAT editor could also be used with other MT systems that produce fragmentary output, if they are incorporated into the TWS. The TWS has in fact been designed to allow such modular growth. It remains unclear how effective the CMAT approach will be in assisting in translation between languages with highly different linguistic structures. We expect to add Japanese as a source language at some point, which should provide some insight into this question.

A serious limitation of the current design is the requirement for each source word to be represented by exactly one of MT, glossary, or dictionary output: while translations of a component may be ambiguous, the boundaries of the component currently may not be. This is a problem, since a word may participate in several possible phrases, which may overlap in arbitrary ways. The result is that the system must arbitrarily give priority to either the MT or the glossary system, and arbitrarily select between overlapping glossary matches, when any of the possibilities may in fact be correct, and so they should all be available to the user as choices. Our solution to this problem will be the use of a chart representation [Kay 1987] for all of the alternatives for all of the components in a sentence. Each alternative in the chart will be indexed by its starting and ending positions in the input. The user will use a new version of the CMAT editor to select among possible phrases that correspond to varying subsequences of words from the source sentence. This could be achieved merely by indicating source phrase/target phrase pairs in menus similar to the current ones. A more ambitious change would be to show the user the whole chart at once, and have the user select which alternatives should be placed into the editor buffer. In either scenario, all possible phrases from the MT system or the glossary would be easily accommodated, without any arbitrary restrictions.

Other enhancements we are considering include displaying different bracketing characters or using different fonts when available for different types of components.

References


Kay, M., 1967. Experiments with a powerful parser. In...