An MAT Tool and Its Effectiveness

Robert Frederking, Dean Grannes, Peter Cousseau, Sergei Nirenburg

Carnegie Mellon University
Center for Machine Translation
Pittsburgh, PA 15213

ABSTRACT

Although automatic machine translation (MT) of unconstrained text is beyond the state of the art today, the need for increased translator productivity is urgent. The PANGLOSS system addresses this dilemma by integrating MT with machine-aided translation (MAT). The main measure of progress in the development of the PANGLOSS system is a gradual increase in the level of automation. The current PANGLOSS MT system typically generates sub-sentence-length units of the target text. Any remaining gaps are treated by lexicon lookup. A mixture of these two kinds of *components* is presented to the user using the CMAT (Component Machine-Aided Translation) editor, which was designed to facilitate the transformation of this output into a high-quality text. An experiment evaluating the utility of the CMAT editor demonstrated its usefulness in this task, and provides useful guidance for further development.

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. 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)[5], has been developed in the framework of the PANGLOSS machine translation project.[1] 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.

PANGLOSS MARK I translates from Spanish into English, although additional source languages are planned. The analyzer used in this configuration is a version of the ULTRA Spanish analyzer from NMSU[2], while generation is carried out by the PENMAN generator from ISI[4]. 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[3].

[1]PANGLOSS is a joint project of the Center for Machine Translation at Carnegie Mellon University (CMU), the Computing Research Laboratory of New Mexico State University (NMSU), and the Information Sciences Institute of the University of Southern California (ISI).

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 (currently, these are 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.

This entire process can 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-editing environment, the CMAT (Component Machine-Aided Translation) editor.

2. The User's View

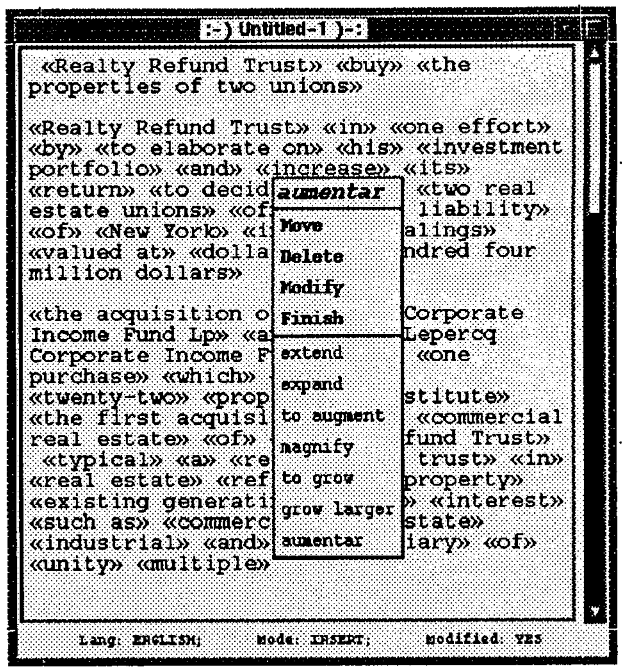
The CMAT editor allows the user to move, replace or delete output text elements, called *components*, with at most two mouse actions. The main user interface tool is a dynamically-changing popup menu available for each component. The ordering of alternate selections in the menus changes as the tool is used, to reflect the most recent user choices.

Suppose the user selects a region of source text by highlighting it and submits it to be machine-translated. The result appears in a target window as a string of components, each surrounded by "≪" and "≫" characters.[2] A mouse click anywhere within a single component brings up a CMAT menu for that component. In Figure 1, the user has clicked on the word "increase". A CMAT menu consists of three regions, each separated by a horizontal line. From top to bottom these are:

- The LABEL region, which contains the word or phrase in the source text that produced this particular component.[3]
- The FUNCTION region, which contains the post-editing **Move, Delete, Modify,** 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. **Modify** pops up a window that allows the user to type in a new alternative (see next bullet). **Finish** removes the component markers, indicating that CMAT editing for this component is finished.[4]
- The ALTERNATIVE region 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.

An additional menu-base editing feature allows the user to change the morphology of a word with a single mouse action (Figure 2). This menu changes verb inflection or the determiner on a noun phrase, stripping any old morphological features before adding the new one.

Using these popup menus, the user can move, replace, modify, 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. 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.



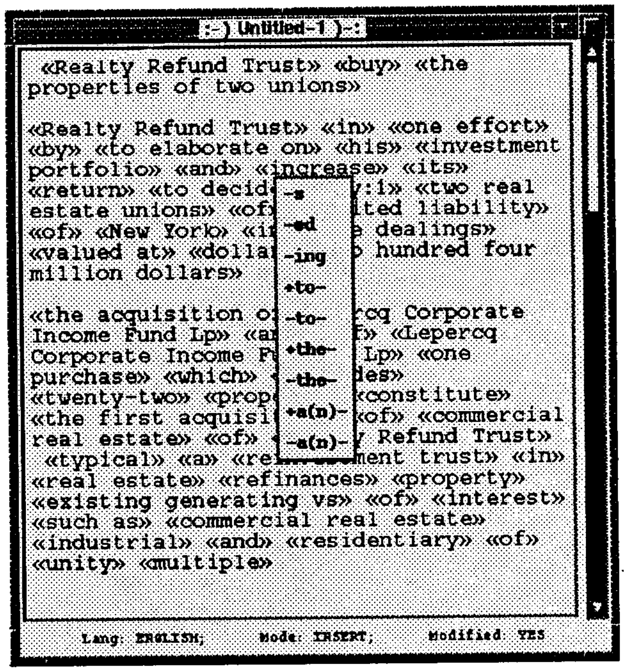
```
«Realty Refund Trust» «buy» «the
properties of two unions»

«Realty Refund Trust» «in» «one effort»
«by» «to elaborate on» «his» «investment
portfolio» «and» «increase» «its»
«return» «to decid            «two real
estate unions» «of             liability»
«of» «New York» «i            alings»
«valued at» «dolla            ndred four
million dollars»

«the acquisition o            Corporate
Income Fund Lp» «a            Lepercq
Corporate Income F             «one
purchase» «which»
«twenty-two» «prop            stitute»
«the first acquisi             «commercial
real estate» «of»             fund Trust»
 «typical» «a» «re             trust» «in»
«real estate» «ref            property»
«existing generati            » «interest»
«such as» «commerc            state»
«industrial» «and»            iary» «of»
«unity» «multiple»
```

Lang: ENGLISH; Mode: INSERT; Modified: YES

Figure 1: A typical CMAT menu



```
«Realty Refund Trust» «buy» «the
properties of two unions»

«Realty Refund Trust» «in» «one effort»
«by» «to elaborate on» «his» «investment
portfolio» «and» «increase» «its»
«return» «to decid       y:1» «two real
estate unions» «of       ited liability»
«of» «New York» «i       e dealings»
«valued at» «dolla       hundred four
million dollars»

«the acquisition o       cq Corporate
Income Fund Lp» «a       f» «Lepercq
Corporate Income F        Lp» «one
purchase» «which»        des»
«twenty-two» «prop       «constitute»
«the first acquisi       «of» «commercial
real estate» «of»       y Refund Trust»
 «typical» «a» «re       ment trust» «in»
«real estate» «refinances» «property»
«existing generating vs» «of» «interest»
«such as» «commercial real estate»
«industrial» «and» «residentiary» «of»
«unity» «multiple»
```

Lang: ENGLISH; Mode: INSERT; Modified: YES

Figure 2: A typical CMAT morphology menu

3. The CMAT Editor

As part of the TWS, the CMAT editor is implemented in Common LISP. It communicates through CLM (the Common LISP-Motif interface)[1] to use Motif widgets inside of the X11 window system.

The CMAT editor views text as a list of components. These

[2]If two components with different internal forms have the same string, it is followed by a colon and an integer.

[3]Note that this information is not always available in noun phrase translation.

[4]The user may also choose to wait and remove all the markers at once, for either a selected region or the whole buffer, using a selection in the TWS's main menu bar or a special keystroke.

components are of three types:

1. **MT-generated strings.** Phrases translated by the MT system are represented simply as the generated target language string, and are not further processed by CMAT.

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 as found in the MT system's lexicons; and finally the identifier `:DICT` and a *dictionary entry list*: a list of target language words found in the machine-readable dictionary.

The CMAT editor uses a knowledge base and a working memory. 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 more than once 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 slots, one slot each for `:GLOSS`, `:MT`, and `:DICT` lists, 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, since these items are more likely to be the correct translations of a source string in our domain.

When a component list is passed to the CMAT editor to be displayed, the latter 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 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, the system must use the actual character string as the index into the working memory, and from there get the index into the knowledge base.[5] The list of alternative translations for the component can then be obtained from the knowledge base structure.

If 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. If 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. The new candidate is also moved to the front of the candidate list as the most likely candidate for future use. When a component is **Modified**, the new alternative entered by the user is stored in the knowledge base, and then treated as if it had just been chosen.

When the component's markers are removed, either singly or *en masse*, the component's pointer in the working memory is removed, but the entry in the knowledge base remains. These are retained in order to provide a summary of the user's preferences, for the frequent case where future translations contain these components. This summary can be saved as a file, which can be loaded into the knowledge base in a later editing session, or analyzed by system developers.

4. The Evaluation of the Tool

In order to evaluate the effectiveness of this tool, we compared editing with the CMAT editor versus editing with just the basic Emacs-like text editor in which it is embedded. We conducted two experiments comparing CMAT and non-CMAT editing efficiency, one using monolinguals and one using translators.

4.1. Experiment I

Method. The monolingual task was to take the output of the MT system and, using as reference an English translation that was previously produced manually, produce the "same" text using either the CMAT editor or the regular editor. The time required for each text-editing session was recorded. Keystrokes and mouse actions were automatically counted by the interface.

As test texts, we used two of the texts from the 1992 DARPA MT evaluation. To shorten total experiment time and provide a reasonable number of sample texts, we broke each text into two halves of roughly equal size, at paragraph breaks, resulting in four text samples.

Two subjects were presented with the samples in the same order. Their use of the CMAT or the plain Emacs editor on

[5]This is due to details of the CLM interface, and is the reason for marking identical components that have different internal data structures with a colon and an integer: otherwise there would be no way to locate the correct associated data structure.

different samples was arranged to provide as much variation as possible in practice effects and acclimatization, so that these could be cancelled out during analysis. A few days later, subjects repeated the procedure, reversing the use or non-use of the CMAT editor. Since practice effects should be more uniform in a simple editing task than in translation (the task is much less intellectually challenging), we felt that texts could be reused if practice effects are taken into account in analysis.

Subjects were instructed to produce a "close paraphrase" of the example translation, since any two translators will produce slightly different correct translations of the same text. Subjects were also instructed not to use the CMAT **Modify** function, since it causes the editor to learn during use, making analysis even harder.

Analysis. Given the above ordering of test runs, one can balance practice effects, subject differences, and text differences simply by normalizing the total editing times for a subject on each run through the texts. That is, if we divide the editing time for each text by the total time for the entire set of texts in the given run, the variation between normalized editing times *between subjects* should reflect variations in the efficiency of editing. For example, in Figure 3, we see that for Session 1, Subject 1 spent a greater fraction of time using CMAT (0.2413) than Subject 2 spent editing it in a regular editor (0.2198), while for Session 2, the fraction of total time was the same with either editor.

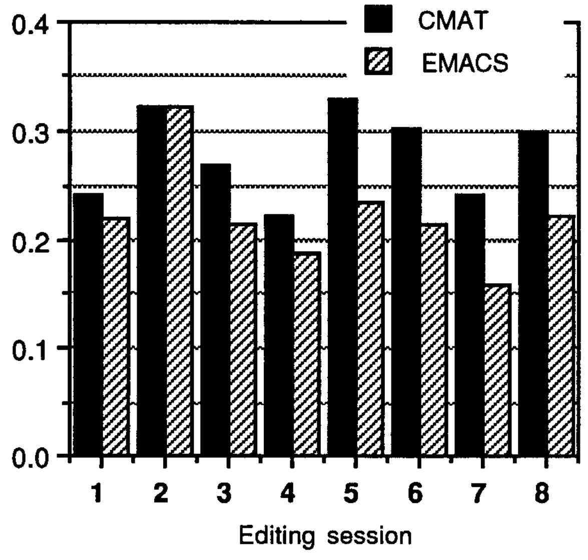


Figure 3: Normalized editing times

From comparing these normalized times, it appears that the CMAT actually slows subjects down. This contradicts the universal subjective impression of all CMAT users that it is quite helpful. It could be the case that the CMAT makes the job *easier* without making it *faster*, but we had the definite impression that it makes translating faster as well as easier. We therefore investigated further.

Normalized keystroke and mouse action counts are shown in Figures 4 and 5. Here we see that while the CMAT editing sessions had 1/2 to 1/3 the number of keystrokes, they had between 2 and 9 times as many mouse operations. This is significant, since mouse actions are slower than keystrokes.

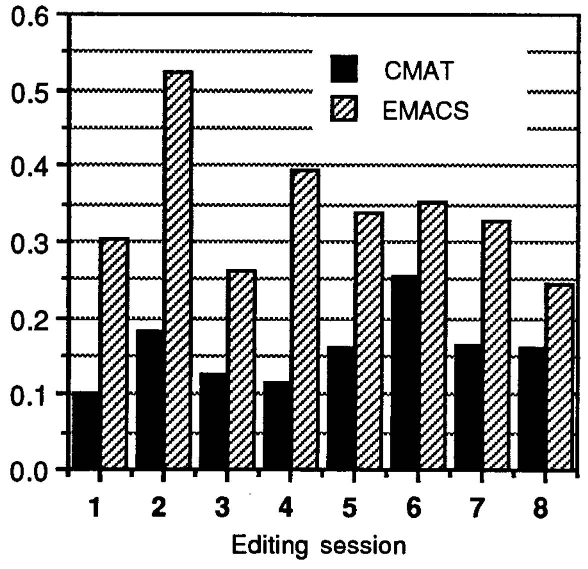


Figure 4: Normalized keystroke counts

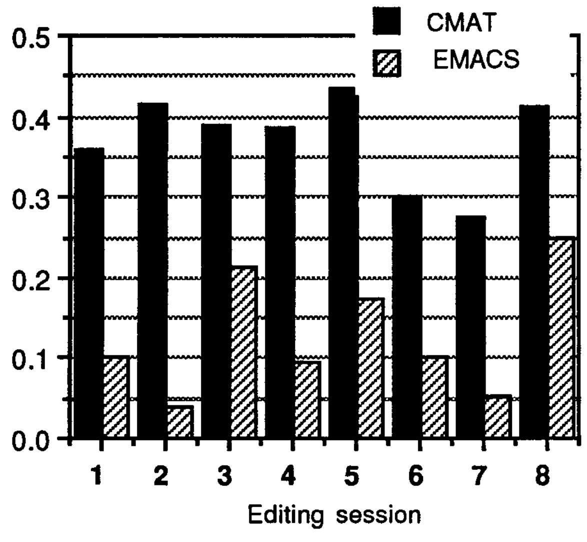


Figure 5: Normalized mouse action counts

Figure 6 shows that Subject 1 has roughly the same speed with or without CMAT, while Subject 2 is about 50% slower editing with the CMAT (top chart). Subject 2 also increases his mouse usage much more during CMAT editing than Subject 1 (bottom chart). Thus all the slow down of the CMAT might be attributable to mouse usage.[6]

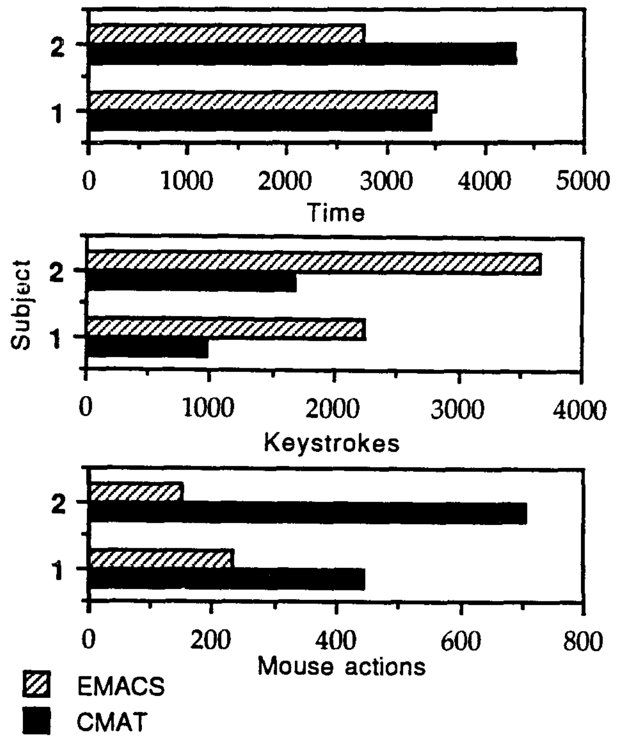


Figure 6: Total times and counts

Experiment I suggested that all the value of the CMAT editor comes from its assistance to translation, since it had either a negligible positive impact or a negative impact on editing times for the two subjects. Emacs is widely regarded as a very good editor, so this hypothesis made sense, but still needed to be tested.

4.2. Experiment II

Method. In this experiment, actual Spanish to English translation was performed. Subjects practiced on a different text until they felt comfortable using the CMAT editor. This time, subjects used the original Spanish text for reference, as is usual when using the TWS. Since translation takes more time, fewer texts were done. Also, each text could only be used once per translator. In order to provide more insight into the exact benefit, if any, of the CMAT editor, times and counts were recorded for both "rough" and "final" drafts of the translations, as judged by the translators. In all other respects, the experiment was conducted like the first one.

[6]It may also be significant that Subject 2 is a touch-typist. His non-CMAT times are faster than Subject 1's, even with roughly 50% more keystrokes.

Analysis. Three translators took part in this test, but one (Subject 4) clearly had not acclimated to using the CMAT, and his data point had to be discarded. He took much longer than anyone else when using the CMAT editor.

Since there had been three subjects, Subject 5 used the CMAT for both texts. Therefore these results had to be analyzed differently than the first experiment: the normalized times for Subject 5 were used to determine the intrinsic difficulty of the two different texts. So if the other subject spent more time on a text than Subject 5, this should have been due to more difficult editing conditions.

As shown in Figure 7, the use of the CMAT editor resulted in an editing time of 89.74% of what would have been expected with Emacs, or in other words a 10.3% speedup compared to non-CMAT editing (top bar). While this number obviously cannot be taken too literally, it does provide an initial indication of a significant speed improvement through the use of the CMAT editor, which should be verified with further testing.

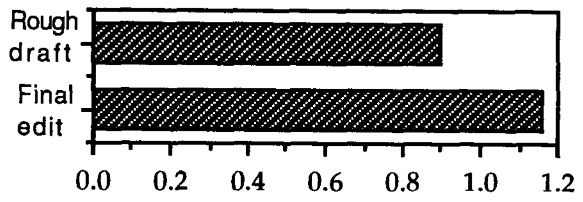


Figure 7: CMAT/Emacs time ratio

In addition, the ratio of the time taken *between* the rough and final drafts by these two subjects, shown in the bottom bar, shows that the CMAT actually slowed editing down a bit in this interval. This makes sense given the earlier results, since essentially all the translation was finished, and only editing remained to be done. So all 10% of the speed-up is attributable to faster translating.

4.3. Discussion

Users of the CMAT editor all indicate that it is helpful in translating. The current test subjects also reported that the CMAT sessions were much easier than the non-CMAT sessions. In fact, one subject tried to avoid taking the second non-CMAT test. Part of this benefit seems to be due to the easy on-line access to relevant dictionary entries. Another benefit is that the brackets provided by the CMAT help the user to segment (and then interpret) the fragmentary English output, and to locate untranslated words. These experiments did not attempt to measure quality differences between test conditions. We intend to carry out further experiments that look for quality enhancement from using the CMAT editor, due to the additional information available to translators, and that measure any trade-off between quality and speed of translation.

In the second experiment, the normalized total-edit time ratios between the two texts for Subject 5 were essentially identical to the rough draft ratios, indicating that this ratio is indeed a good indicator of the relative difficulty of the two passages.

It is interesting to note that Subject 4, whose data point had to be thrown out because his CMAT times were twice the length of his non-CMAT times, corresponds closely to the level of familiarity our translators had with the CMAT editor in the first MT evaluation in 1992. An important part of our preparation for the 1993 MT evaluation will be training the test subjects in the most efficient use of our tools.

5. Conclusion and Future Work

The CMAT editor, in conjunction with often fragmentary MT and word-for-word translation, allows the translator to produce high-quality translations more quickly and easily than the simple combination of a text editor and an online dictionary. It will remain a crucial module in PANGLOSS until the MT system reaches the point of translating full sentences on a regular basis.

These experiments provide initial evidence that the CMAT editor is indeed effective, and have been very useful in pointing out areas for improvement:

- The current CMAT design requires the use of the mouse. Since mouse actions are often slower than keystrokes, we will provide keybindings for *all* CMAT commands, including viewing and selecting alternative translations. This should not be technically difficult.
- The users need to be taught the most effective strategies for using the CMAT, such as only using the mouse if they are fast with it, and generally not using the CMAT after their rough draft is finished.
- Currently the CMAT menu often does not contain the correct answer, due to the low-quality of the online dictionary. This dictionary is currently being replaced, and we expect the coverage to be much improved for the next MT evaluation.

References

1. Bäcker, A., C. Beilken, T. Berlage, A. Genau, M. Spenke, 1992. CLM – A Language Binding for Common Lisp and OSF/Motif: User Guide and Reference Manual, Version 2.1, Technical report, German National Research Center for Computer Science.
2. Farwell, D., Y. Wilks, 1990. ULTRA: a Multi-lingual Machine Translator. Memoranda in Computing and Cognitive Science MCCS-90-202, Computing Research Laboratory, New Mexico State University, Las Cruces, NM, USA.
3. Kugler, M., G. Heyer, R. Kese, B. von Kleist-Retzow, G. Winkelmann, 1991. The Translator's Workbench: An Environment for Multi-Lingual Text Processing and Translation. In Proceedings of MT Summit III, Washington, DC.
4. Mann, W., 1983. An Overview of the Penman Text Generation System. In Proceedings of the Third AAAI Conference (261-265). Also available as USC/Information Sciences Institute Research Report RR-83-114.
5. Nirenburg, S., P. Shell, A. Cohen, P. Cousseau, D. Grannes, C. McNeilly, 1992. Multi-purpose Development and Operation Environments for Natural Language Applications, In Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP-92), Trento, Italy.