For Want of a Nail: Machine Translation and the Word Processor

Larry G. Childs

FOR WANT OF A NAIL:
MACHINE TRANSLATION AND THE WORD PROCESSOR

Larry G. Childs
ALPNET

At last October's annual conference of the American Translator's Association in Seattle, Charles Teubner of the Princeton Technical Translation Center in New Jersey explained to a very interested audience how he uses Systran, one of the oldest and largest machine translation companies. His source documents are written in Mass-11, which is a word processing system especially suited for representing scientific formulas and equations. He said that in order to send his documents through the Systran system, he first has to strip out the formatting commands by hand, and then after receiving his translations back from Systran, he must put the formatting commands back in manually as well.

Based on my own experience with computer translation, I found his situation to be quite indicative of the state of much commercial machine translation today. There are several interesting points which can be drawn from his story.

First, machine translation (MT) is being used today in a commercial setting. This proves that despite the numerous linguistic problems and difficulties, machine translation can be a viable, productive tool. And although I got the impression that Mr. Teubner does not use MT on large quantities of material, there are many other businesses and organizations today which do use the products of the various MT companies successfully on very high volumes of text.

Second, source texts which are used in commercial machine translation systems are often created in machine-readable format. Typical source texts for MT include user manuals for everything from computers to tractors, and nowadays these are almost all created on word processors. This fact has helped to alleviate the considerable problems associated with first converting source text into machine-readable form in order to use it in machine translation.

The third point which can be drawn from this story, and which I wish to elaborate on in this paper, is that even when source texts are created in machine-readable form to begin with, there are still problems involved in getting the texts to work for machine translation, or at least to work well. In other words, it is not always easy to get machine-readable source text into a format that can be used by a machine translation system.

Mr. Teubner is obviously keen enough on the idea of machine translation to take the time to adapt his texts by manually removing and replacing formatting commands in order to use an MT system. But this manual method is hardly an economical way to do machine translation, and for large volumes of text it becomes far too time consuming to be feasible at all.

Systran might be criticized for failing to provide some sort of automated solution to this problem, but they cannot be singled out for blame. It has been my experience that none of the other commercial MT companies have done much better. It has only been within the last year or two that the company for which I work, ALPNET, has seriously addressed the issue.

I think MT companies have been slow to respond to this need because they have failed to appreciate the importance and extent of the problem. Systran and most of the other MT companies came into being before the virtual explosion in the number of different word processing systems over the past several years, and consequently their systems were not originally designed to deal with a variety of word processing formats. As the need has arisen, the various MT companies have underestimated the difficulty in adapting the various word processing formats for use by machine translation systems, assuming that it was a relatively simple logistical problem. However, if not dealt with adequately, this problem can destroy the economic viability of a commercial translation system. To paraphrase an old saying: "For want of usable source text, the machine translation system was lost."

What exactly are formatting commands, and why are they so difficult to deal with? Every document, i.e., computer file, that is created with the use of a word processor contains not only the text that the author typed in, but also a whole host of typesetting information, known as markup or formatting commands, which indicate such things as margin width, tab settings, spacing, centering, justification, pitch, font, table of contents and index entries, bolding, italics, and underlining, to name some of the most common ones. Depending on the word processing system, some commands are placed automatically in the file as default values. Others are put in by the author as he or she wishes to modify the format of the document.

These commands are generally character strings, just like any other regular word in the file, but they have special meaning to the word processing system and are generally not visible in the printout of the word processing document. Within the computer file itself, however, these commands can occur virtually anywhere. Some types of commands occur in blocks between paragraphs or at the beginning of the file; others are interspersed with the regular linguistic text of

the file. And depending on the type of word processing system used, it may not be immediately obvious to the typist that some of these commands are being mixed in with the regular text.

The placement of markup commands is illustrated in Appendices 1 and 2. Appendix 1 shows the printout of a text created by the WordPerfect word processing system. Appendix 2 shows the same file with the markup commands visible.

When these markup commands occur within sentences, it wreaks havoc with the output of a machine translation system. The commands are interpreted as "words" in the sentence by the computer, but of course, since these "words" have no grammatical function, they only serve to confuse the computer analysis. The resulting translation is often so garbled, that the human revisor has to retranslate the entire sentence manually, thus negating the advantages of using the computer for translation.

In order to use word processing files at all in machine translation, the markup commands must first be removed from the linguistic text of the source document. And then, if the translation is in any way to reflect the same format as the source document, the markup commands must be put back in the translated text. It is possible, of course, to do all this by hand, but the only really economical method is to do it automatically.

Automatically removing the markup commands can be fairly trivial; many word processors can do this as a standard feature. The difficulty for machine translation comes in automatically putting them back into their corresponding positions in the translation. Because the word order of the translation is often quite different from that of the source, it is difficult even to define exactly what "corresponding position" means.

Let me illustrate the problem by discussing a simple algorithm for dealing with markup commands. Before we remove the commands from the source text, let us mark their position by associating each command with the word next to it. In our algorithm, we will arbitrarily associate each command with the word to its right. Then in the translation, each command is put back in the text to the left of the translation of the word with which it was associated.

This method works well for certain commands, such as "[FtnOpt]" (footnote option) in Appendix 2, whose exact position in the sentence is not critical. The [FtnOpt] command merely specifies the printed format of subsequent footnotes and can appear anywhere in the text before the first footnote whose format is to be specified. In this algorithm, the [FtnOpt] command in Appendix 2 would be

associated with the word "with" and would then appear to the left of the translation of "with" in the target text.

For other types of commands, where exact position in the sentence is more important, the results of this algorithm can be disastrous. Take highlighting commands, for example. These are commands such as "bold", "underline" and "italic", which indicate how a word or a phrase should be highlighted. These commands come in pairs: a begin highlight, and an end highlight command.

Figure 1 shows an example of the underline commands in WordPerfect. Everything between the begin highlight ([U]) and the end highlight ([u]) is to be underlined. Figure 2 shows the results of applying our algorithm to these commands when translating into Spanish. Instead of "arte moderno" being underlined, only "moderno" is.

Fig. 1 He studies [U] modern art [u] in Madrid.

Fig. 2 Estudia arte [U] moderno [u] en Madrid.

We may be tempted to say that this algorithm doesn't work merely because of the arbitrariness of associating each command with the word to the right. It is obvious that end highlight commands stand in a relationship with the word to the left, not to the right. In our example, [u] is really associated with "moderno", not "en".

However, if we refine our algortihm to associate end highlight commands with the word to the left, the results, shown in figure 3, are even worse. We end up with the end highlight command before the begin highlight.

Fig. 3 Estudia arte [u] [U] moderno en Madrid.

Actually, all algorithms based solely on associating formatting commands with adjacent words are insufficient. The best way to handle highlighting commands is to mark each word between a begin and end highlight command pair as belonging to that particular highlighted phrase, and then removing the highlight commands from the source text. In the translation, a begin highlight command is inserted between each pair of words where the left word is not marked for highlighting, but the right word is. The end highlight command is inserted between each word pair where the left word is marked and the right word is not.

This algorithm also works well when a highlighted phrase is split into two or more non-contiguous groups in the translation. Take, for example, the single English word "open" which is highlighted in the sentence in figure 4. When translating into German, this single word is split into two non-contiguous words. See figure 5. In this case, not

only do the original commands have to be replaced, but new commands have to be generated to surround each split-off element.

Fig. 4 [U] Open [u] the window.

Fig. 5 [U] Machen [u] Sie das Fenster [U] auf [u] .

It is even possible to highlight only part of a word, as in figure 6. However, unless the computer somehow maintains a correspondence of morphemes between each source and target word, which would be very difficult, if not impossible, to do, there is not a perfect solution to this problem. Perhaps the best approach in this case would be to replace the end highlight command at the end of the word from which it was taken. Conversely, if a begin highlight command is in the middle of a source word, it could be put back before the beginning of the translation of the word. This would leave the entire word highlighted in the target, but manual post-editing could easily restore the original format.

Fig. 6 [U]under[u]line

There are still other types of markup commands which are problematic. Sometimes the command itself contains text which should be translated. The WordPerfect footnote command, for example, contains the entire text of the footnote, yet in the internal file structure, the footnote text is placed within the footnote marker inside the original sentence. See Appendix 2. In this case, the command must be removed in order to translate the original sentence, but the text within the command still has to be presented to the machine translation sytem at some point in order to be translated itself.

Other commands, even though not needing to be translated, fulfill some linguistic function and should be left in the sentence when it is translated. IBM Corporation's ISIL markup system ("Information Structure Identification Language"), for example, has commands known as "variables" which stand for nouns, usually product names. At some point in the processing of the document, the variables are replaced by the nouns that they stand for, but during translation (and IBM uses machine translation to translate many of its documents), they are generally still variables. These commands typically function as subjects or objects in the sentence and must be considered as nouns by the machine translation system.

The same sort of command exists in word processing systems which have a "form letter" or "merge" capability. Appendix 3 shows an example of a Microsoft Word form letter. The words and phrases in the letter which are enclosed by guillemets (« ») and begin with lower case letters are special

variables, called "field names", and are linguistically significant. (The other commands, viz. IF, ELSE and ENDIF, are discussed later in this paper.) Form letters also have an accompanying "data document" which contains the "values" for these variables. In this case, the data document consists of a list of customers' "name"s, plus the "product" that each customer is interested in along with the "retail" price. The form letter and the data document are then "merged" when printing; i.e., a different, personalized letter is printed out for each customer in the data document.

In some word processing systems, these linguistically significant commands can stand for more than just nouns. ISIL, for example, also contains a whole set of markup commands which stand for different kinds of punctuation marks, such as colons and quotation marks, and which, for various formatting reasons, are typically used in a document rather than the actual punctuation. A machine translation system, of course, must recognize each different command and treat it as the punctuation mark for which it stands. There may be dozens of different commands that have to be treated individually for each word processing system that is handled, which can make the automatic handling of markup commands extremely complicated.

Finally, let me give one more example of a class of markup commands, which to my knowledge has not yet been encountered enough in commercial machine translation to become a problem, but which does illustrate some of the nastier possibilities which markup commands can present. Some word processing systems have a series of commands which essentially turn the text file into a sort of computer program, and can change the final form of a document quite drastically depending on the value of certain "variables". Appendix 3 illustrates the Microsoft Word, "IF", "ELSE", and "ENDIF" commands. In the form letter shown, the text of the final paragraph reads quite differently depending on the value of the "retail" variable. In a machine translation system which must grammatically parse each sentence, this sort of thing can be quite a problem indeed.

In conclusion, word processing presents an interesting challenge to the machine translation industry. In the past, the problems of handling markup commands have been overlooked or dismissed as fairly simple logistical details. In fact, dealing adequately with markup commands is critical to the economic usefulness of machine translation, and some of the problems involved in dealing with them are not trivial at all.

APPENDIX 1 -- WORDPERFECT FILE PRINTOUT

Sometimes a phrase is underlined to emphasize a point. At other times boldface type is used. Combining highlights lends extra emphasis to what has been written.

Other commands, such as the footnote format command, can be interspersed with the text as well. And although footnotes are printed at the bottom of the page, in internal WordPerfect file structure, they are actually stored with the footnote marker[1] in the text.

> Another common practice is to indent entire paragraphs in order to set them off from the rest of the text. In this parargraph, the pitch has also been changed from pica to elite, and the right margin has been justified, which highlights the paragraph even more.

---

[1]This is the body of the actual footnote.

APPENDIX 2 -- WORDPERFECT FILE INTERNAL STRUCTURE

[Pg Lnth:32,20] [Top Mar:20] [Margin Set:10,66] Sometimes a phrase [U] is underlined [u] to emphasize a point. At other times [B] boldface type is used. [b] Combining highlights lends [B] [U] extra emphasis [b] [u] to what has been written.

Other commands, such as the footnote format command, can be interspersed [FtnOpt] with the text as well. And although footnotes are printed at the bottom of the page, in internal WordPerfect file structure, they are actually stored with the footnote marker [Note:Foot,1;[Note #]This is the body of the actual footnote.] in the text.

[Font Change:12,1] [Rt Just On] [->Indent<-] Another common practice is to indent entire paragraphs in order to set them off from the rest of the text. In this parargraph, the pitch has also been changed from pica to elite, and the right margin has been justified, which highlights the paragraph even more.

APPENDIX 3 -- MICROSOFT WORD FORM LETTER

Dear «name»,

Thank you for your request concerning our «product». The «product» sells for «retail» plus 5% tax and handling. If you wish to order the «product», please remit a total of: «retail + retail * .05».

We would also be glad to send you our catalog at «IF retail > 10» no extra charge «ELSE» a cost of ony $2.00 «ENDIF».

Sincerely,