[*Terminologie et Traduction*, no.1, 1986]

IMPROVING MACHINE TRANSLATION QUALITY OR ELIMINATING THE "INVISIBLE IDIOT"

Jeanne Homer,
World Translation Center, La Jolla, California

Preface

During this presentation, I will be describing my sense of machine translation developments over the next few years independent of specific translation systems. However, when I refer to Systran, or use the pronoun WE during the presentation, I am speaking about the system developed and maintained in La Jolla, California at LATSEC and at World Translation Center and not about other versions of Systran that now exist.

Introduction

Machine translation has come a long way since Dr Peter Toma founded LATSEC in the 1960s. It has proved to be an acceptable and cost-effective alternative to manual translation among users with high-volume, technical translation needs.

As we all know, however, significant improvements in machine translation are needed to increase its applicability to a broader base of users and document types.

Factors Impacting Future Machine Translation Slide

As the next slide indicates, three factors will influence the future of machine translation. First, improving international economic and political factors will increase the demand for the translation of documents. Secondly, improved translation techniques will result in higher quality machine translation systems. And third, continued advances in technology will facilitate the implementation of the above translation techniques and generally increase the efficiency of automated translation systems. I am not a technologist and thus do not intend to concentrate on the technology aspect. However, as a linguist and systems analyst, I will address the topic of improved translation techniques.

Improved Translation Techniques Slide

For reasons of practicality, I will focus on the three areas which will have the most significant impact on translation quality over the next few years. These areas, or as I call them, translation techniques are:

- increased modularity of the translation system,
- implementation of semantic capabilities, and
- streamlining of dictionary coding.

Components of machine translation quality Slide

Before discussing these areas, I want to identify the criteria for measuring translation quality.

The quality of a translation system, or even a human-based translation service, can be evaluated using the following 5 criteria;

- Accuracy
- Speed
- Ease to use
- Adaptability/Flexibility
- Cost

Let me describe what I mean by each of these.

Accuracy

A general definition of accuracy is that a translated document be understandable and reflect the intent of the original document. More specifically however, accuracy addresses the questions:

- Is the translation grammatically correct? and

- Is the translation semantically accurate?

Speed

Speed of translation includes both:

- How long does it take to translate a document? and

- How long does it take to enter information into the dictionary?

Ease of Use

Ease of use includes such things as:

- Must the source language text be in any particular format?

- How easy is it to access and interface with the translation system? For example, will the translation system accept word processing documents produced by multiple vendors? And will the output be in a format which can interface with other document processing systems such as electronic mail and word processing?

- Can the user influence improvements to the system through providing feedback on the dictionary and programs?

Adaptability/Flexibility

The criteria for evaluating adaptability and flexibility are:

- Will the translation system accept user-specific terminology?

- How easy is it to develop a new language pair? and

- Can the system take advantage of new technology?

Cost

Cost includes:

- How much does it cost to have a document translated? and
- How much does it cost to enter new terminology into the dictionary?

Now I want to talk about each of the three translation techniques.

Modularity

The first factor I'd like to discuss is modularity. The dictionary defines MODULE as "a separable component, frequently one that is interchangeable with others, for assembly into units of differing...function".

I'd like to give some examples of why modularity is important by explaining how we have been using this concept in dictionaries and programs and how we might make future use of this technique.

Dictionary Modularity

First let's look at the use of modularity as it relates to dictionaries.

Systran dictionaries include not only the word and its meaning, but also a variety of grammatical information about each word. In the past, a single dictionary was created and maintained for every pair of languages translated by Systran. Thus, we had a dictionary for the English-French system and another dictionary for the English-Spanish system. If a new target language was added, still another separate dictionary was created. New words and expressions needed to be redundantly entered into each dictionary. The problems with this are obvious - redundant coding consumes time and increases the potential for error and inconsistencies.

Today we have only one dictionary for each source language. We call this a "multi-target" dictionary. Each "multi-target" source dictionary contains key pertinent information about each source language word as well as the associated meanings for all target languages. Thus all new source language vocabulary is entered only once into the "multi-target" source dictionary, and all systems based on that source language are enhanced. The same, of course, is true for modifications to the dictionary entries of source language words. So for example, if an error in the coding of a German word is discovered while we are translating a German-French text, the resulting dictionary change is made once, and the improvement is automatically reflected in the German-Spanish, German-English and German-Italian systems.

In the future, I expect to see the development of a single target language dictionary as well. In other words, the German-Spanish system and the English-Spanish system would both utilize the same Spanish target dictionary. Each Spanish word and its corresponding grammatical information would be entered only once into the Spanish target language dictionary.

Then any source language which has a word with that particular meaning would simply code a reference to that Spanish word and not re-think and re-enter all the information pertaining to the word into the "multi-target" source dictionary.

There could be another evolutionary step beyond this in dictionary modularity. It would be possible to have only one dictionary for each language, (i.e. as opposed to a separate source and separate target dictionary for each language). Such a dictionary would provide all information, both source and target, about a given word. For each such word, there would be a cross reference to the associated target words in the related dictionaries.

Program modularity

Now I'd like to discuss the use of modularity in the translation programs.

As you know, each translation system consists of analyzing a source language and then synthesizing into a target language. Early systems, some of which are still in use, do not strictly separate the analysis and synthesis phases. Typically this lack of separation means that the analysis is done knowing what the target language is and the synthesis is done knowing what the source language is. If there were only a single language pair to be translated, this approach would be acceptable. The reality, of course, is that there are multiple language pair systems to develop and maintain. The approach of "integrated" analysis and synthesis creates two problems. First, it creates a system that is difficult to enhance and maintain because the analysis and synthesis routines are tailored to specific languages and the code intermixed. Second, language pairs with the same source language tend to have redundant analysis routines, which further complicates maintenance.

Program: analysis modularity

Systran's source language syntactic analysis programs are modular in function. Thus, the same syntactic analysis is performed regardless of the target language.

Historically however, semantic analysis (i.e. lexical disambiguation) has been performed during the synthesis phase. Let me clarify this with an example.

Many words have different meanings depending upon context. The English word for GRADE, for example, can mean both "a degree of rank", as in GRADE AA EGGS and "the rate of ascent or descent of a slope", as in A 5% GRADE. One method for resolving semantic ambiguity is through "lexical" routines that perform context dependent analysis. Thus, if each definition of GRADE requires a different meaning in the target language, a lexical routine must be written to determine the usage in a particular sentence and then assign the correct meaning in the target language.

In the past, and even today in some of the Systran systems, context-dependent lexical routines are written for each language pair. In other words, the same ambiguity in meaning for GRADE was resolved in each language pair system where English was the source language.

Some years ago, WTC introduced the concept of multi-target lexical routines. In these routines, the ambiguity in meaning is resolved at the source language level, which is where the ambiguity exists. The correct definition of GRADE is determined during analysis. Synthesis then uses this information to identify the proper target language meaning.

Program: synthesis modularity

While analysis can and should be independent of the target language, certain parts, but not all, of the synthesis process are, in fact, dependent upon the source language. In the past, however, most translation systems had distinct synthesis routines for each language pair even when the target languages were the same.

Recently, WTC carried the concept of modularity one step further, in creating synthesis programs which are highly independent of a particular source language. For example, not only are all conjugation tables and routines for forming the plural "unaware" of the original language of the text, but routines for determining tense, mood, voice, choice of article and word order are also source-language independent.

Let me describe two examples of synthesis that can be made independent of the source language.

> First, it is a rule of Dutch grammar that an adjective preceding a singular neuter noun is not inflected when it is not preceded by an article. This rule is not duplicated in the English-Dutch and French-Dutch synthesis programs. Rather this synthesis rule is in a special "Dutch only" synthesis program which is independent of the source language.
>
> Second, another synthesis function is to create the correct order of words in the target language. This year, we plan to deal with the particular issue of adverb placement in English. In the example on the slide, you can see that the adverb follows the predicate in French and is separated from the predicate by the direct object in German.
>
> However in English, it must appear immediately before the predicate. Since the position of the adverb is dependent only upon the idiosyncrasies of that particular English adverb and not on the source language, this routine will be written only once and can be used to improve the translation for all systems in which English is the target language.

Of course it's still necessary to have some source language dependent synthesis programs, for particular language pairs such as French-English or German-Spanish. However, an increasing number of rules will be entered in a purely target synthesis program.

I hope these examples have helped show how modular dictionaries and programs can be used in a variety of ways to improve Systran. Let's summarize the impact of modularity on the components of quality.

Modularity also has a positive impact on accuracy. The elimination of repetition will reduce dictionary coding and programming errors and will ensure that improvements to any module will be reflected in, and thus improve, all systems using that module. Let me emphasize that users of one language pair, say German-English, might think they do not benefit from this concept, but, in fact, their system will improve with every modification made to German analysis (whatever the target language) and to English synthesis (no matter what the source language is). In a non-modular system the changes would have to be made several times, or, worse, not made at all.

The most significant impact will be on the system's adaptability to new language pairs. Adding a new target module to an existing source language or a new source language to an existing target language requires considerably less effort than starting from scratch and creating a new combination of existing language pairs involves even less effort.

Since the analysis and synthesis functions are so clearly separated, modularity will also have a favorable impact on cost due to reduced dictionary and program maintenance and increased efficiency. Of course, any improvements in accuracy will result in a reduced post editing effort.

Semantics

The second translation technique I want to discuss is the use of semantics. I believe that Systran will achieve its full potential only when it is able to recognize the semantic content, as well as the syntax of text. One approach to achieving semantic recognition is through the use of semantic categories. A semantic category characterizes a group of words of similar linguistic type. Some examples of semantic categories are FOOD PRODUCT, STRESS, RIVER and ABSORB. The concept is that the meaning of a word in a particular context, as well as its grammatical functioning, can be determined by identifying the semantic qualities of words found in the same context.

Let me illustrate the problem of semantic recognition with an example. This slide illustrates two of the meanings for the word CASE. Specifically, CASE can mean either a CONTAINER or INSTANCE OF SICKNESS.

In this instance, because the object of OF (i.e. MEASLES) is in the semantic category DISEASE, the meaning is not CONTAINER, but an INSTANCE OF SICKNESS.

In our system, there are nearly 500 semantic categories, organized in a hierarchy or tree structure. The bottom-most points, or terminal nodes of the structure represent very specific categories. As one goes up the tree, the categories on the nodes become more general.

For example, when the word SHOVEL is coded with the semantic category EQCON (construction equipment), it is automatically assigned the higher categories DEV (device), PHYSUB (any concrete tangible thing), INAN (not animate), PHENOM (item which can be perceived by the senses) and finally THING, one of six basic categories.

Semantic recognition is not only of obvious value in determining the meaning of a word during analysis, but can also be useful in other aspects of the analysis such as determining syntactic relationships, as well as during synthesis.

Let me explain this further with some examples:

- Analysis currently relies almost exclusively on the syntactic properties of the words in a sentence. Thus a sentence can be parsed accurately according to rules of syntax, but in fact be incorrect because semantic knowledge was not used. In the French example at the top of the slide, the word "extrêmes" could syntactically modify "pluie", "neige", and "températures". Only by using semantic knowledge do we know that there is no such thing as extreme rain or snow, and that "extrêmes" must therefore modify only "temperatures".
- Semantic information can be useful during analysis in establishing appropriate relationships, such as between adjective/modifier, subject/predicate, verb/object, potential members of enumeration, and apposition. In the sentence on the bottom of the slide, the word "augmentation", since it follows the conjunction "et", is identified as being in a parallel construction with some word to the left of the word "et". But is it with "équilibre", "perte", "pression" or "viscosité"? There are some syntactic clues, but ultimately the decision must be made on the basis of the semantic knowledge that "perte" and "augmentation" (or "loss" and "increase" in English) are in the same category.
- As I mentioned earlier, another important area for use of semantics is in the selection of meanings. Lexical routines and CLSs (which are mini lexical routines in the LS or "expression" dictionary) must rely on the semantic properties of words in order to resolve ambiguities and ensure the correct translation.
- The two examples on the next slide illustrate this point. The principal meaning for the French verb TIRER is "to draw". However, when the object of TIRER belongs to the category EQWAR or "equipment of war", then the meaning of TIRER should be FIRE, and not DRAW. In the second example, we see the German verb STOPFEN, which means "to stuff" in English. But when the object is a garment (semantic category GARMNT), the meaning should be "to mend".
- Finally, synthesis will also achieve better translations by relying on semantic properties. Choosing the best translation for an adnominal genitive construction or solving the difficult problem of what article to use are just two examples of the potential usefulness of this feature in synthesis.

Of all our language pairs, our Russian-English system currently has the most sophisticated semantic capacity and is now achieving accuracy rates of approximately 90%. It is my opinion that full-scale incorporation of semantic capacities into all of our systems will result in similar increases in accuracy and thus reduced post editing effort.

Let's now look at the impact of semantics on the five components of quality.

As you can see by the check marks, the use of semantic capabilities will have a substantial impact on the accuracy of the translation, by improving both the grammar and the choice of meanings.

Improvements in accuracy reduce post-editing effort and thus cost.

Streamlining of dictionary coding

The third translation technique I want to discuss is dictionary coding. As I'm sure you all know, dictionary coding is the process of entering words, their meanings and related grammatical information into the translation system's dictionary.

To appreciate how dictionary coding can be improved, allow me to summarize how it has been done. Typically, a bilingual coder fills out a coding form for each word. He or she first writes the stem of the word, using all capital letters and adding digits for any accent marks or for any letters which do not occur in English. Then the coder enters the conjugation or declension pattern it belongs to, the homograph type, the part of speech, and various syntactic and semantic codes which pertain to the word. Then the meaning of the word, which may have several stems (see the example for ACQUERIR) and its related inflection pattern and syntactic codes are entered. This information is then keyed by clerical personnel, verified by a dictionary edit program and finally merged into the stem dictionary.

As you know, we also have an expression dictionary for each language. This dictionary not only contains expressions, but also allows for very sophisticated CLS rules which assign meanings based on the function of the word and context of a particular sentence - mini lexical routines. In some instances, this dictionary has been used in order to solve linguistic issues on a meaning by meaning basis, which could better be solved by addressing the issue as a whole.

Let's now look at how to streamline the coding process. 1) Make it easier for the coder to enter the necessary information, 2) Avoid repetition, and 3) Where possible decrease the need for LS coding. Let's look at these one at a time:

First, in order to make it easier for the coder to enter information into the dictionaries:

a. Coding should be done at a terminal, using an interactive system with immediate automatic verification of codes. This would consolidate the three steps mentioned above.

b. Both the source and target meanings should be entered in a natural format, using upper and lower case letters as well as appropriate accents. Instead of multiple stems, only the basic form of the word should be entered, e.g. infinitive for verbs, the singular of the noun and the masculine singular positive form of adjective. This is most natural for a coder and is also the format of entries in non-computer dictionaries.

c. When coding a stem entry, the first information a coder should enter after the source language word, is the pertinent semantic categories. (I believe this method is already being used by some Systran systems.) The interactive system can then respond immediately by filling in as much information as can be learned from the category entered.

Let's take the word DOCTOR, for example. If the coder enters the semantic category PROF (for PROFESSION), the screen can immediately fill in:
- primary part of speech - NOUN
- secondary part of speech - TITLE
- the syntactic codes HU (human), ANIM (animate), CON (concrete noun), and CT (count noun), as well as the most common inflection type for nouns. These codes can then be verified by the coder.

Some people believe that coding can be streamlined by not coding as much information with each word, but this must necessarily lead to poorer quality translations. The better approach is to simplify the process so that more effort can be directed toward choosing the correct meaning.

The second streamlining technique is to avoid repetition. As I mentioned when discussing modularity, we've already eliminated repetition of coding by creating multi-target source language dictionaries. If this same concept is continued by creating one target language dictionary for each target language, or as I suggested, one all-encompassing dictionary for each language, then repetitive effort will be entirely eliminated. An additional advantage to this proposed approach is that it will not be necessary for coders of the basic dictionaries to know more than one language. He or she will just need to know the grammar of his own language. Only the person who links the source and target meanings will need to know both languages.

The third aspect is reducing the dependence on LS coding for solving linguistic issues.

For example, we've just completed a project dealing with the treatment of German separable-prefix verbs. Until this issue was treated systematically, every single German separable verb had to have one or more CLSs written in order to translate the verb correctly when it occurred in its separated form. As a result of the recent programming effort, however, we were able to reduce the size of the German LS dictionary by 2000 entries.

This same type of systematic treatment of French adnominal genitives has resulted in a corresponding decrease in French LS coding.

Increasing reliance on semantic capabilities will similarly be used to eliminate the need for such extensive LS coding of English noun phrases.

Going back to our chart of the components of quality, we can see that the streamlining of dictionary coding will have an impact on all areas.

The greatest impact, of course, will be in making the system much easier to use and in increasing the speed of entering information into the dictionary. Automatic code generation will improve the accuracy of the dictionary, and that will have a direct impact on the accuracy of the translation.

Flexibility is increased due to the ease of adding the vocabulary of a new language pair into the dictionary. Finally, cost is reduced as a result of all of the above factors.

Technology

As I mentioned earlier, technological advancements will also play an important role in the future of machine translation. Some of the advances that will have an impact on machine translation include voice recognition, improved optical character readers, high capacity, lower cost communication networks, international document content and interface standards, continued price/performance improvement for processors and storage, smaller storage devices with larger capacities, and artificial intelligence tools.

Now I'd like to summarize the impact of technological advancements on machine translation quality. In many ways improved technology is a prerequisite for achieving the improved translation techniques we discussed this afternoon. In total they will impact each of the "component of quality" criteria. However they will have the greatest impact on speed, ease of use and cost.

Conclusion

I have prepared a final slide summarizing the impacts of both technology and the improved translation techniques on the five components of quality.

It paints an encouraging picture. The bottom line is that machine translation systems of the future will be much easier to access, will require little if any human intervention, will support an increasing number of languages, will produce high quality translations for technical documents and acceptable translations for non-technical material and finally the cost of translations will be markedly lower than manual translation services for documents of all sizes.

I've intentionally limited the "future" during this presentation to the next few years. However, the future beyond the "next few years" will be every bit as exciting and will produce quantum leaps in machine translation through such things as voice-operated, miniaturized translation systems that can be carried by an individual and sophisticated artificial-intelligence-based translation systems. This however is a topic for another day.