COMMUNICATION &LANGUAGE at work
Issue no. 1 | Summer 2012
**ICT Tools and Professional Language**

# Machine Translation Tools – Tools of the Translator's Trade

Peter Kastberg
(Translated by Thomas Buch Andersson)

*(pp. 34 - 45)*

http://ojs.statsbiblioteket.dk/index.php/claw/article/view/7238



# Machine Translation Tools

*Tools of the Translator's Trade*

Peter Kastberg
Assoc. Prof., Ph.D.
*Dept. of Business Communication, School of Business and Social Sciences, Aarhus University*

**Abstract**

*In this article three of the more common types of translation tools are presented, discussed and critically evaluated. The types of translation tools dealt with in this article are: Fully Automated Machine Translation (or FAMT), Human Aided Machine Translation (or HAMT) and Machine Aided Human Translation (or MAHT). The strengths and weaknesses of the different types of tools are discussed and evaluated by means of a number of examples. The article aims at two things: at presenting a sort of state of the art of what is commonly referred to as "machine translation" as well as at providing the reader with a sound basis for considering what translation tool (if any) is the most appropriate in order to meet his or her specific translation needs.*

## Translation tools

Translation tools are generally understood as software helping the translator to translate a written text from one natural language (the source language) into a text in another natural language (the target language).

From this definition follows two important limitations: Firstly, translation tools are developed with the purpose of translating written texts between various languages, in contrast to, for example, speech recognition systems. Speech recognition systems typically "translate" within the same language from one medium (speech) to another (text). Secondly, it is a case of the translation of texts from a *natural language*. Thus, we are entering the framework of what is referred to as "natural language processing" or NLP.

### The different types of translation tools

The existing translation tools can roughly be divided into three types:

- Fully Automated Machine Translation (FAMT)
- Human Aided Machine Translation (HAMT)
- Machine Aided Human Translation (MAHT)

Like with most classifications, this one is not entirely unproblematic either and it can be difficult to precisely define the boundaries between the different translation tools. The criteria reasoning this tripartition are centered on two aspects: One aspect is *the translation tool's degree of automation,* i.e. the degree to which the machine/software independently conducts the actual translation. The other is the question regarding *who controls the translation process*, i.e. who – the software or the translator – controls which decisions are made in the process.

I will elaborate on these aspects in my examination of the characteristics of each of the three translation tools below. In my examination the programs will be presented as abstract models, and I will therefore not make use of existing, concrete programs.

# Fully Automated Machine Translation (FAMT)

Fully automated machine translation (or simply Machine Translation or MT) is understood as software capable of conducting a flawless translation between natural languages independent of human interference or help. The human role has been reduced to simply loading the desired text into the computer.

Ever since machine translation-related research truly took off in the 1930s – primarily because technological advances made it possible – the dream has been to develop a machine – today it would be referred to as software – capable of translating without any other human interference than the actual order to translate a given text from language A to language B. Directed by the Allies' successful attempts at breaking the German military codes during World War II, attempts were made to develop FAMT-programs from the middle of the 1940s. The basic idea behind these programs was that natural languages were comparable to codes, and codes could be broken. Success was achieved in the middle of the 1950s when a team of scientists from Georgetown University, USA, had a machine successfully translate a number of sentences from Russian to English. Naturally, this success led many universities to establish their own development centers for machine translation. However, as early as the middle of the 1960s, enthusiasm stalled. During a large-scaled American analysis dubbed ALPAC (Automatic Language Processing Advisory Committee), voices within the area expressed doubts as to the possibility of ever being able to develop a fully automated translation program. Some researchers bluntly said that fully automated machine translation was impossible. Consequently, research and development of FAMT-programs was either assigned a lower priority or completely terminated. Practically, this meant that FAMT 'hibernated' throughout the 1970s; and it was not before the development within information technology gained serious momentum in the 1980s (especially in regard to storage capacity and processing speed), that FAMT again was taken seriously. Through the 1990s research and development in fully automatic translation regained its popularity, one of the reasons being the scale of the potential marked from the software developer's

perspective. In addition, availability of information technology increased greatly with the pc, and use of information technology is now a natural part of everyday life.

Based on this information, I will now review the three types of currently available FAMT-programs. During the review I will comment on the characteristics of each type.

## The different types of FAMT-programs

FAMT-programs can, like the whole of translation tools (section 1.1), roughly be subdivided into three types:

- The Direct MT-Model
- The Transfer Model
- The Interlingua Model

**The Direct MT-model** was the first model developed. It is intended to provide a translation between two pre-determined languages, i.e. it is not possible to load any text from any language and receive a translation in any desired language. The translation process in the Direct MT-model consists, in principle, of a local morphological analysis of the source language text (i.e. an analysis of the words' forms, functions etc.) and an algorithm capable of applying these results to the morphology of the target language and to a bilingual dictionary capable of recognizing and substituting words from the source language with words from the target language. Thus the Direct MT-model builds on the idea that a translation is a question of two things: Firstly, to conduct a local morphological analysis, and secondly, to recognize words in the source language and then translate those words into words of the target language. The simplicity of the idea is appealing, but in reality it is not that simple.

The Direct MT-model knows from its morphological analysis that a noun phrase like [the man] consists of the noun [man] and the determiner [the]. On that basis it can, for example, translate [the man] into the definite form of the Danish noun [mand], which is [mand] with the definite ending [-en], i.e. [manden]. This is perfectly okay per se, but as the morphological analysis is *local*, it cannot analyze words in their context. Practically, this entails that the word order in the source text becomes the word order of the translation. Let us look at a German example:

[ich bin es]

In the Direct MT-model this could be:

[I am it]

From a morphological perspective, i.e. in respect to the form of the words, the suggestion from the Direct MT-model is perfectly acceptable. The German words [ich] and [bin] have been correctly translated to the English [I] and [am], and not to e.g. [you] and [were] or something completely different. For the German word [es] the English

equivalent is [it], and thus also perfectly acceptable from a morphological perspective. The problem is that we in English simply say something different:

[it is me]

In other words, in English the equivalent of the German [ich bin es] is constructed differently.

The other problem with the Direct MT-model is a lexical problem. By analyzing locally the program – popularly speaking – translates word-by-word, and, in doing so, is not designed to analyze and translate the words from their context. This is especially problematic when the words to be translated either have more than one meaning or whose meaning, for some reason, is difficult to define. Let us look at an example where we imagine that the Direct MT-program is asked to translate a text in which the following word is included:

[cell]

The word [cell] can in English denote a wide range of phenomena:

1. Cell in a monastery
2. Cell in a prison
3. Cell in a living organism
4. Cell as a group in an organization or political movement
5. Cell in a beehive

A number of more professional/technical metaphors wherein [cell] is part of are presented below:

6. Cell in the context “fuel cell”
7. Cell in the context “solar cell”

Etc.

Which of the different denotations of [cell] is actually meant in the source text is, naturally, crucial to how the word should be translated. The human translator has the advantage that he or she can read and understand the context of the word when making a choice between the various denotations. From the context the human translator is, in the vast majority of cases, capable of determining which of the denotations the author had in mind when he or she wrote [cell]. This advantage is not available to the Direct MT-program – qua its local analysis. Therefore, the program will often translate the “primary meaning” defined by the developer of the program. If we, for example, imagine that the “primary meaning” of [cell] has been defined by the programmer as being a “cell in a beehive”, it can give rise to some confusion on the readers’ behalf if the author of the source text means “cell in a living organism”. Practically, this problem occurs if the program, for example, is set to translate the English word [cell] into Italian.

Here “cell in a beehive” is called “cella” (the 5th denotation above), while “cell in a living organism” is called “cellula” (the 3th denotation above).

The second type of FAMT-programs is **the Transfer model**, which, in many ways, is an evolution of the Direct MT-model. Similar to the Direct-MT model, the actual translation is a matter of; the program conducting an analysis of the source language, an algorithm guiding the results to the target language, while a bilingual dictionary is replacing the words from the source language with words from the target language. Like the Direct MT-model, the Transfer model can only translate between a pre-determined pair of languages.

However, in contrast to the Direct MT-model, the Transfer model’s analysis is not merely local and morphological, but *regional* and *grammatical*. In other words, the Transfer model conducts a more comprehensive analysis of the source text in two areas: It does not just analyze single words, but also groups, or strings, of words that belong together. To enable an analysis of the words in context, the program does not just contain a morphological description of the individual words, but also a grammatical description in terms of how words in the source language and target language, respectively, are put together correctly. This is exactly where the force of the Transfer model is when compared to the Direct MT-model.

Let us consider an example. If we ask a Transfer-program to translate a group of words that in their context produce, what we in a grammatical sense know as, an attribution:

[the artificial flower]

Considering what we know about the Direct MT-model, we can determine that the Direct MT-model would not be able to see how these words are connected and instead translate them word-by-word to e.g. French:

[la artificielle fleur]

However, this construction is not typical (correct) in French, where the adjective [artificielle] in an attribution normally succeeds the noun [fleur]. Since the Transfer-program analyses regionally, the built-in grammar ensures that the phrase is not just translated correctly in terms of morphology, but in terms of word order as well:

[la fleur artificielle]

Due to the fact that the Transfer model analyses regionally, its built-in grammar is so advanced that it, in French, knows to put the adjective after the noun in an attribution. However, not all French adjectives follow the noun in attributions like this one. For example, there is a group of basic adjectives like “grand”, “petit” and “jolie” that should precede the noun. The Transfer model can overcome this problem by combining the previously mentioned grammatical rule with its built-in dictionary. Now the

combination could look like this: In French, the adjective follows the noun with the exceptions of the adjectives "grand", "petit", and "jolie" where the adjective, in these cases, precedes the noun. Thus the Transfer model is based on a rather advanced algorithm, which – as far as we know – resembles the mental process of a human translator when he or she is translating an attribution from one language to the other.

Nevertheless, even though a Transfer-program takes us further with regard to grammatical precision than a Direct MT-program, the Transfer-program has its limitations as well. One of which is that we rarely express ourselves in grammatical constructions limited to a single attribution. The attribution is typically part of a sentence, which in turn is typically part of a longer text, which again is part of something bigger, that is, the entire communication context. The Transfer model has not been designed to take these parameters, which go beyond what we could refer to as sentence grammar, into account.

**The Interlingua model** is the last type of FAMT-program I will approach, and also the most ambitious of the fully automatic translation models we know today. In principle, the translation process of the Interlingua model looks like this: The text of the source language is fed to the program, and analyzed *globally* and *semantically* (i.e. for meaning). The analytical results are then transferred to a semantic code, known as Interlingua, which is designed to reproduce the contents of the source text. In contrast to the two previously mentioned models that focus on the source text's *expressions* (e.g. morphology and grammatical rules), the Interlingua model focuses on the source text's *content*. In other words, it is by taking point of departure in the contents of the source text that the Interlingua model is considered such a breakthrough. This means that the algorithm of the Interlingua model is so advanced that it can transfer the Interlingua to a natural target language. This transaction happens through a so-called generator, and thus it is said that the program generates a translation.

If you compare this to how a human translates, the idea of basing the translation on the contents of source text is evident. When a human translator conducts a translation it is exactly the contents or the meaning that is translated. The morphology or grammar is – so to say – simply how the meaning is presented.

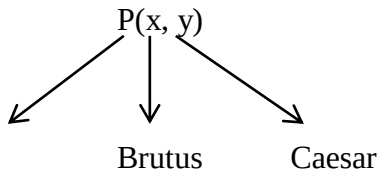
Thus the goal of the Interlingua model is deliberately to mimic the human mental process that is believed to occur when he or she translates. Thereby, with the knowledge we have of languages today, the fundamental problem in the first attempts of creating fully automatic translation machines becomes apparent. The problem is that natural languages are simply not built on some fundamental logic code that only needs to be broken to enable translation (if this was the case we would have had fully automatic translation programs built-in to our Office solutions long ago). It is to a certain degree possible to convert single words or single strings of words into logic relations capable of being translated by a machine, however, it is not possible to convert languages as

such into logic codes, nor can we convert the way humans interact with the language into logic codes.

As an example, let us look at the sentence "Brutus kills Caesar". When converted to a logic language, such a sentence will typically be reproduced like this:

P(x, y)

If we add words to it, it can be disintegrated to:

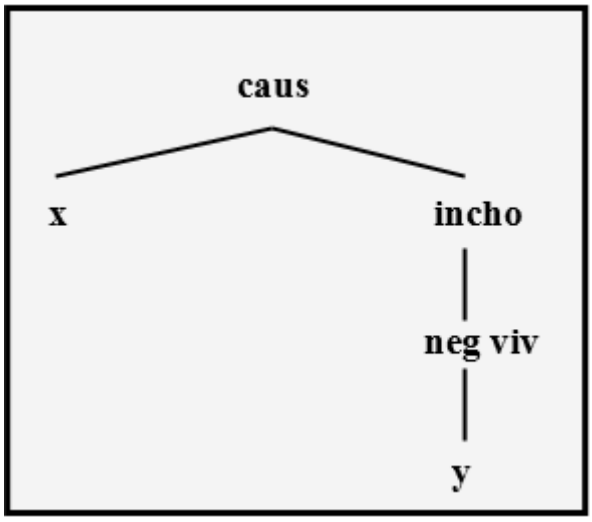



However, "P(x, y)" is also a logic code for "Marie bakes bread", "Michael makes coffee" etc. I am deliberately putting things on the edge here, but it is a rather good illustration of how far a very abstract and logic representation of our language takes us.

As mentioned, the Interlingua model takes another direction and is based on the content. If we, for example, type the sentence

[Brutus kills Caesar]

into the Interlingua-program, the program would transfer the meaning of the sentence to an Interlingua which could look like this:



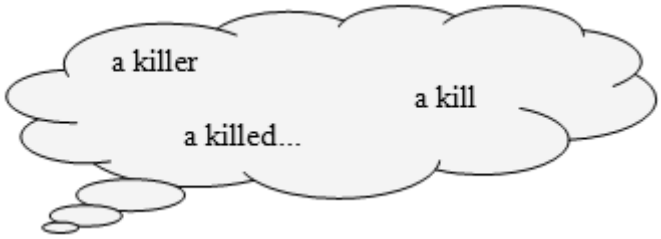

Or presented in a more reader-friendly way:

[Y causes the circumstance that X does not live]

Here, the sentence has been separated into single parts of meaning while making the relations between them evident. This abstract recollection of meaning, and the relations

in between its parts, now has to be transformed, by the program, to a relation of meaning in the target language, by using the so-called generator.

However, to enable a sentence from a natural language to be transformed to an Interlingua, it is essential for all words to be defined. In order to understand this process it is necessary to look at the bigger picture. If we isolate one of the words from the above sentence – like the verb "to kill" – and think about which elements of meaning produce our mental model of that word, it could look like this:




[to kill]

In other words, "to kill" involves (at least) three fundamental elements in its meaning – which are: a killer, a killed, and a kill - , and it is such fundamental elements of meaning that forms the cornerstones of the Interlingua. However, at the same time, it is the fundamental elements of meaning that cause the Interlingua model's big problem. This is due to the fact that the Interlingua is based on the idea that it should be possible to isolate all fundamental elements of meaning, and that all these fundamental elements of meaning are universal, i.e. that they are similar in all countries and languages. Thus if the fundamental elements of meaning are not universal, or cannot be described universally, the Interlingua cannot function inherently as an Interlingua. For an expression like "to kill" it is probable that it covers (at least) these three fundamental elements of meaning. That is, in every homicide – independent of language and culture – there will be a killer, someone who is killed, and thereby a kill. But how are less concrete expressions handled? Expressions like "spirit", "fortune", "honor" or simply "insolence". These words are not just abstract, but also words that in different cultures express different things, and/or can have a different (social) value. In other words, the Interlingua model is by no means unproblematic, as it demands de facto that a universal language of expressions is developed or abstracted; something that could be compared to a kind of Esperanto of expressions.

## Concluding remarks on FAMT-programs

If we were to conclude today where machine translation stands, so to speak, we would have to say that; *despite* extensive research and development efforts in Europe, USA and Japan, *despite* the increasing processing power of computers, and *despite* the increasing "intelligence" of programs, it still has not been possible to develop a FAMT-program superior to the good human translator. However, it needs to be said that every software generation of programs will improve – no doubt about it – and that with the

next generations of neural programs and – on a somewhat longer term – the development of artificial intelligence, we will see fully automated translation programs of a completely different caliber than we know today.

Although it has not been proved as easy to develop a fully automated translation program as was believed in the 1930s, research and development has by no means stalled, the prospects are simply too attractive: If – or perhaps when – the development of a fully automatic translation machine, capable of translating as well as its human counterpart, is successful, it would mean:

- that written communication across languages and cultures would no longer be hindered by a lack of foreign language skills
- a considerable reduction in time. All it would take to translate any text on any subject of any length would be the time it takes to press a button or click the mouse
- a considerable reduction in costs, as the human translator, who is expensive in education and salary, is no longer needed

Spurred on by the problems encountered in the development of the fully automatic translation machine, translation software, where the machine no longer understands the entire translation process but instead where man and machine are cooperating in the translation process, were developed in parallel. Programs of this sort are the above-mentioned HAMT- and MAHT-programs, that I will now approach (section 3 and 4, respectively)

# Human Aided Machine Translation (HAMT)

HAMT is understood as software developed for the machine to translate what it can, in the way it can. The human role can be compared to that of a consultant or an editor, i.e. that the translator corrects or modifies what, in the machine's translation suggestions, is unacceptable to him or her.

- SYSTRAN is an example of a HAMT-program.

In an HAMT-program the human translator can, in principle, take on his/her consultant- or editor-role *before*, *during* or *after* the machine has provided its translation. Naturally, these three phases of processing can be combined in several ways.

If the processing takes place *before* the machine has been set to translate, it is a case of pre-editing. Pre-editing is when the translator adapts the source text to enable the program to decode the text.

Such a modification could, for example, be to adapt all the sentences in the source text to have the same word order, or a word order typical to that of the target language. Pre-

editing thus makes it possible to avoid the word order problem of the target text that we saw in the example with the Direct MT-model above. Another case of pre-editing could be to re-write the source text into a so-called "controlled language", i.e. a language that has been strongly conventionalized, that is stereotypical in its sentence structure and has a minimum of stylistic variation. Such a "controlled language" would be perceived by most people as mechanical or unappealing, however, it is easier to design a program to analyze such a language than to analyze a natural language. An example of a "controlled language" is the so-called "air speak" which is used to communicate in international aviation.

If the processing takes place *during* the translation process, it is a case of interactive editing. Typically, the program would ask the human translator a number of questions, which the program is not designed to answer itself, regarding the solution to concrete translation problems. As an example, the program could ask the translator to consider which of the 7 meanings of the word "cell" we discussed above, is meant in the source text.

If the processing takes place *after* the machine has produced its translation, it is a case of post-editing. Here the translator should correct the machine's translation suggestions in the same way many teachers today correct their student's translations/assignments, or in the way linguists and proofreaders today review and correct the texts of others.

In other words, if one translates with the help of an HAMT-program, it is not possible to avoid being actively involved in the translation process. This is also the case if the translator only post-edits. The conscientious or critical translator/proofreader/editor will, when stumbling upon something that has been translated seemingly incorrect by the computer, have to compare the computer's target text with the source text. Only by comparing the text of the target language with the text of the source language can the translator check if the computer has provided an acceptable and correct translation. Thus, he or she does not avoid being deeply engaged with the source text.

# Machine Aided Human Translation (MAHT)

MAHT is understood as software that, in one way or the other, helps the human translator whenever he or she asks for it. In its simplest form, MAHT-software can for instance be spell check and grammar check. The slightly more advanced MAHT-software includes electronic dictionaries, terminology databases etc. Today, the most advanced form of MAHT-software is Translation Memories (TM). This also includes "Computer Aided Translation" or CAT.

MAHT-tools, like spell check, grammar check and databases, are imbedded in most Office solutions, as internet resources or to be had on a CD. I will not elaborate on this type of MAHT-tools as they are developed for the general PC-user and not particularly

for the translator. Instead I will elaborate on a MAHT-tool developed specifically for use during the translation process and thus I will, in the following, address the essential features of translation memories.

- Translator's Workbench from TRADOS is an example of a translation memory.

A translation memory is software that functions by storing what previously has been translated within the program. To be able to translate in such a program requires quite a lot of preceding work for the human translator. Let us take a closer look at the translation process when using a translation memory:

The source text is typed into the program, where its format is adapted to that of the translation program, this process is called alignment. The aligned document then has to be exported to a text file, which subsequently is imported to the actual translation memory. Once this preceding work is completed the translation process can be initiated.

The translator will now, on his or her PC-screen, be presented with the source text which, through the program, has been divided into Translation Units (or TUs). Such translation units are typically sentences. Each time the translator has translated a translation unit, the program saves the original translation unit from the source text and its corresponding translation. As the translation process progresses, an increasing number of "pairs" will have been saved and it is the collection of these that makes up the translation memory.

When the program recognizes a translation unit that has been translated previously, it notifies the translator and simultaneously shows how he or she has previously translated an identical translation unit. All the translator then needs to do is to copy these suggestions. Also when no 100% identical precedents in the memory are available, most programs can recognize parts of the available translation units and on this basis suggest translations. In such scenarios the program typically indicates that it is a case of a "fuzzy match", i.e. it shows; that it is not a completely identical precedent, and the degree to which this "fuzzy match" resembles the actual translation unit (e.g. is the match 50% or 90%). It is then up to the translator to evaluate which parts of the suggestion he or she can use.

In addition to the actual translation memory, such programs also consist of a number of search functions, enabling a direct search for a word or translation unit in the memory or an attached (terminological) database.

Practically, all this is reflected on the translator's PC-screen, which, during a translation that is assisted by a translation memory, has been separated into a number of sections. As a minimum there will be four types of sections:

- a section consisting of the TU currently being translated by the translator
- a section indicating if any previously translated TUs resembles the current TU.

and derived from those

- a section indicating "match" or the degree of "fuzzy match" between the current TU and a previously translated TU

together with

- a number of sections enabling searches in the memory, databases etc.

Translation Memories are especially relevant when the frequency of repetition is high. An example could be when the translator has to translate texts that are linguistically similar in their structure. This is typically the cases of manuals or instructions, where directions or requests are continuously expressed in a similar way. Another example is when a great number of pages dealing with the same topic have to be translated, which is the case with much scientific and technical literature. By using a translation memory, the translator is ensured a high degree of consistency in word choices and formulations where appropriate. Such consistency in linguistic expressions is, additionally, an important parameter for companies with a formulated language policy. Another considerable aspect, in regard to the use of translation memories, is that the translation memory deals with all repetitions and thereby the monotonous part of the translation process, thus, in principle, giving the human translator more time either for the more creative, challenging and thereby more exciting aspects of the translation process, or simply to translate more. An additional advantage is that there is not, in principle, a limit to the size of such a translation memory. Today, sufficient storage capacity will always be available, either on the hard drive, a server or in the cloud. For the translator this could ideally mean that her or she could, eventually, build translation memories so extensive that they, in the long run, will be capable of supplementing or perhaps even completely substituting external literature references, dictionaries, encyclopedias etc.

# Author

**Peter Kastberg**

Assoc. Prof., Ph.D., Dept. of Business Communication, Aarhus University, Business and Social Sciences

Peter Kastberg holds a Ph.D. in applied linguistics (technical communication). He is the coordinator of the research network on "Sociology of Knowledge" at the Aarhus University, Denmark. Among his current research interests count: mediation of specialized knowledge across knowledge asymmetries, the ontogenesis of knowledge in institutional contexts, as well as public understanding of science and research. Peter Kastberg's teaching areas include: corporate communication, academic rhetoric, communication theory, translation theory, and knowledge communication.

**Contact:**

pk@asb.dk

*Aarhus University, Business and Social Sciences*
*Fuglesangs Allé 4, 8210 Aarhus V*
*Denmark*